Advances in Intelligent Systems and Computing 582

Wojciech Zamojski
Jacek Mazurkiewicz
Jarosław Sugier
Tomasz Walkowiak
Janusz Kacprzyk *Editors*

Advances in Dependability Engineering of Complex Systems

Proceedings of the Twelfth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, July 2–6, 2017, Brunów, Poland

Springer

Advances in Intelligent Systems and Computing

Volume 582

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within “Advances in Intelligent Systems and Computing” are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

More information about this series at http://www.springer.com/series/11156

Wojciech Zamojski · Jacek Mazurkiewicz
Jarosław Sugier · Tomasz Walkowiak
Janusz Kacprzyk
Editors

Advances in Dependability Engineering of Complex Systems

Proceedings of the Twelfth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, July 2–6, 2017, Brunów, Poland

Springer

Editors

Wojciech Zamojski
Department of Computer Engineering
Wrocław University of Science
and Technology
Wroclaw
Poland

Jacek Mazurkiewicz
Department of Computer Engineering
Wrocław University of Science
and Technology
Wroclaw
Poland

Jarosław Sugier
Department of Computer Engineering
Wrocław University of Science
and Technology
Wroclaw
Poland

Tomasz Walkowiak
Department of Computer Engineering
Wrocław University of Science
and Technology
Wroclaw
Poland

Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
Warsaw
Poland

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-59414-9 ISBN 978-3-319-59415-6 (eBook)
DOI 10.1007/978-3-319-59415-6

Library of Congress Control Number: 2017940846

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume presents proceedings of the Twelfth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX which took place in the Brunów Palace in Poland from 2nd to 6th July 2017.

The volume appears in the series "Advances in Intelligent Systems and Computing" (AISC) published by Springer Nature, one of the largest and most prestigious scientific publishers, in the series which is one of the fastest growing book series in their programme. The AISC is meant to include various high-quality and timely publications, primarily conference proceedings of relevant conference, congresses and symposia but also monographs, on the theory, applications and implementations of broadly perceived modern intelligent systems and intelligent computing, in their modern understanding, i.e. including tools and techniques of artificial intelligence (AI), computational intelligence (CI)–which includes neural networks, fuzzy systems, evolutionary computing, as well as hybrid approaches that synergistically combine these areas–but also topics such as multiagent systems, social intelligence, ambient intelligence, Web intelligence, computational neuroscience, artificial life, virtual worlds and societies, cognitive science and systems, perception and vision, DNA and immune-based systems, self-organizing and adaptive systems, e-learning and teaching, human-centred and human-centric computing, autonomous robotics, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, various issues related to "big data", security and trust management, to just mention a few. These areas are at the forefront of science and technology, and have been found useful and powerful in a wide variety of disciplines such as engineering, natural sciences, computer, computation and information sciences, ICT, economics, business, e-commerce, environment, health care, life science and social sciences. The AISC book series is submitted for indexing in ISI Conference Proceedings Citation Index (now run by Clarivate), EI Compendex, DBLP, SCOPUS, Google Scholar and SpringerLink, and many other indexing services around the world.

DepCoS-RELCOMEX is an annual conference series organized since 2006 at the Faculty of Electronics, Wrocław University of Science and Technology, formerly by Institute of Computer Engineering, Control and Robotics (CECR) and now by Department of Computer Engineering. Its idea came from the heritage of the other two cycles of events: RELCOMEX (1977–89) and Microcomputer School (1985–95) which were organized by the Institute of Engineering Cybernetics (the previous name of CECR) under the leadership of Prof. Wojciech Zamojski, still the DepCoS chairman, so this year we can celebrate the 40th anniversary of its origins. In this volume of "Advances in Intelligent Systems and Computing", we would like to present results of studies on selected problems of complex systems and their dependability. Effects of the previous DepCoS events were published (in chronological order) by IEEE Computer Society (2006–09), by Wrocław University of Technology Publishing House (2010–12) and presently by Springer in "Advances in Intelligent Systems and Computing" volumes no. 97 (2011), 170 (2012), 224 (2013), 286 (2014), 365 (2015) and 479 (2016).

Dependability is the contemporary answer to new challenges in reliability evaluation of complex systems. Dependability approach in theory and engineering of complex systems (not only computer systems and networks) is based on multidisciplinary attitude to system theory, technology and maintenance of the systems working in real (and very often unfriendly) environments. Dependability concentrates on efficient realization of tasks, services and jobs by a system considered as a unity of technical, information and human assets, in contrast to "classical" reliability which is more restrained to analysis of technical resources (components and structures built from them). Such a transformation has shaped natural evolution in topical range of subsequent DepCoS conferences which can be seen over the recent years. This edition additionally hosted the 7th CrISS-DESSERT Workshop devoted particularly to the challenges and solutions in analysis and assurance of critical infrastructure and computer (software and programmable logic-based) system safety and cybersecurity.

The Programme Committee of the 12th International DepCoS-RELCOMEX Conference, its organizers and the editors of these proceedings would like to gratefully acknowledge participation of all reviewers who helped to refine contents of this volume and evaluated conference submissions. Our thanks go to, in alphabetic order, Andrzej Białas, Ilona Bluemke, Eugene Brezhniev, Dariusz Caban, Frank Coolen, Manuel Gil Perez, Zbigniew Huzar, Igor Kabashkin, Vyacheslav Kharchenko, Leszek Kotulski, Alexey Lastovetsky, Jan Magott, István Majzik, Jacek Mazurkiewicz, Marek Młyńczak, Yiannis Papadopoulos, Oksana Pomorova, Krzysztof Sacha, Rafał Scherer, Mirosław Siergiejczyk, Janusz Sosnowski, Jarosław Sugier, Victor Toporkov, Tomasz Walkowiak, Irina Yatskiv, Wojciech Zamojski and Włodzimierz Zuberek.

Thanking all the authors who have chosen DepCoS as the publication platform for their research, we would like to express our hope that their papers will help in further developments in design and analysis of engineering aspects of complex systems, being a valuable source material for scientists, researchers, practitioners and students who work in these areas.

The Editors

Twelfth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX

organized by
Department of Computer Engineering,
Wrocław University of Science and Technology
Brunów Palace, Poland, 2–6 July 2017

Programme Committee

Wojciech Zamojski (Chairman)	Wrocław University of Science and Technology, Poland
Ali Al-Dahoud	Al-Zaytoonah University, Amman, Jordan
George Anders	University of Toronto, Canada
Włodzimierz M. Barański	Wrocław University of Science and Technology, Poland
Andrzej Białas	Institute of Innovative Technologies EMAG, Katowice, Poland
Ilona Bluemke	Warsaw University of Technology, Poland
Eugene Brezhniev	National Aerospace University “KhAI”, Kharkiv, Ukraine
Dariusz Caban	Wrocław University of Science and Technology, Poland
Krzysztof Cios	Virginia Commonwealth University, Richmond, USA
Frank Coolen	Durham University, UK
Mieczysław Drabowski	Cracow University of Technology, Poland
Francesco Flammini	University of Naples “Federico II”, Napoli, Italy
Manuel Gill Perez	University of Murcia, Spain

Zbigniew Huzar	Wrocław University of Science and Technology, Poland
Igor Kabashkin	Transport and Telecommunication Institute, Riga, Latvia
Janusz Kacprzyk	Polish Academy of Sciences, Warsaw, Poland
Vyacheslav S. Kharchenko	National Aerospace University “KhAI”, Kharkiv, Ukraine
Mieczysław M. Kokar	Northeastern University, Boston, USA
Krzysztof Kołowrocki	Gdynia Maritime University, Poland
Leszek Kotulski	AGH University of Science and Technology, Krakow, Poland
Henryk Krawczyk	Gdansk University of Technology, Poland
Alexey Lastovetsky	University College Dublin, Ireland
Marek Litwin	ITS Polska, Warsaw, Poland
Jan Magott	Wrocław University of Science and Technology, Poland
Istvan Majzik	Budapest University of Technology and Economics, Hungary
Jacek Mazurkiewicz	Wrocław University of Science and Technology, Poland
Marek Młyńczak	Wroclaw University of Science and Technology, Poland
Yiannis Papadopoulos	Hull University, UK
Oksana Pomorova	Khmelnitsky National University, Ukraine
Ewaryst Rafajłowicz	Wrocław University of Science and Technology, Poland
Krzysztof Sacha	Warsaw University of Technology, Poland
Rafał Scherer	Częstochowa University of Technology, Poland
Mirosław Siergiejczyk	Warsaw University of Technology, Poland
Czesław Smutnicki	Wrocław University of Science and Technology, Poland
Janusz Sosnowski	Warsaw University of Technology, Poland
Jarosław Sugier	Wrocław University of Science and Technology, Poland
Victor Toporkov	Moscow Power Engineering Institute (Technical University), Russia
Tomasz Walkowiak	Wrocław University of Science and Technology, Poland
Max Walter	Siemens, Germany
Bernd E. Wolfinger	University of Hamburg, Germany
Leonid G. Voskressensky	Peoples’ Friendship University, Moscow, Russia

Min Xie	City University of Hong Kong, Hong Kong SAR, China
Irina Yatskiv	Transport and Telecommunication Institute, Riga, Latvia
Włodzimierz Zuberek	Memorial University, St.John's, Canada

Organizing Committee

Honorary Chairman

Wojciech Zamojski

Members

Włodzimierz M. Barański
Jacek Mazurkiewicz
Jarosław Sugier
Tomasz Walkowiak

Seventh CrISS-DESSERT Workshop

Critical Infrastructure Security and Safety (CrISS) -
Dependable Systems, Services & Technologies (DESSERT)

The CrISS-DESSERT Workshop evolved from the conference *Dependable Systems, Services & Technologies* DESSERT 2006–2016 (www.dessertcon.com). The 6th CrISS-DESSERT took place in Chernivtsi, Ukraine, 21–22 May 2016. In 2017, the 7th CrISS-DESSERT Workshop was held in the frameworks of the 12th Conference on Dependability and Complex Systems DepCoS-RELCOMEX.

The mission of the Workshop was to discuss challenges and solutions related to analysis and assurance of critical infrastructure and computer (software and programmable logic-based) system safety and cybersecurity. In particular, its focus was chosen in order to address:

- interplay and interdependencies of system of systems (telecommunication, smart grid, intelligent transportation system, etc.) and the current problems in providing its safety, security, reliability, quality of services, etc.;
- roles played by IT (SW, HW, FPGA)-based systems as the mandatory part of each infrastructure, thus turning distinct infrastructures into a complex cyber-physical system (system of systems) with emergent and cooperative behaviour, uncertainties, etc.;
- resource-effective IT-based approaches to safe and sustainable development.

The CrISS Workshop examined modelling, development, integration, verification, diagnostics and maintenance of computer and communications systems and infrastructures for safety-, mission- and business-critical applications.

Main Topics

The main topics on the workshop agenda included the following:

- Formal methods for critical IT infrastructures and systems development and verification
- Vulnerability analysis and intrusion-tolerant systems
- Evolving infrastructures and self-systems

- Dependability and resilience of Web-, cloud- and IoT-based IT infrastructures
- Safety of human–machine interfaces and systems including cooperative HMI&S
- Functional/system safety perspective of intelligent transport systems (ITS)
- Information & data modelling in ITS context
- IT infrastructures for pre- and post-accident monitoring of critical objects
- Safety- and assurance-case methodologies, techniques and tools
- Smart grid safety, reliability and security
- Power saving in IT infrastructures, data centres and computing clusters

Workshop Panel Discussion

Infrastructure and industrial systems safety and security: challenges, monitoring and assurance case-based solutions.

Workshop Chair

Vyacheslav Kharchenko	National Aerospace University KhAI, Centre for Safety Infrastructure Oriented Research and Analysis, Ukraine

Co-chairs

Todor Tagarev	Institute of Information and Communication Technologies, Bulgaria (TBC)
Nikos Bardis	Hellenic Military Academy, Greece

Industry Partner

RPC Radiy, Ukraine

Contents

Detection of Network Attacks Using Hybrid ARIMA-GARCH Model

Tomasz Andrysiak(✉), Łukasz Saganowski, Mirosław Maszewski, and Adam Marchewka

Faculty of Telecommunications and Electrical Engineering,
Institute of Telecommunications,
University of Technology and Life Sciences (UTP),
ul. Kaliskiego 7, 85-789 Bydgoszcz, Poland
{andrys, luksag, mmasz, adimar}@utp.edu.pl

Abstract. In this article, an attempt to solve the problem of attacks (anomalies) detection in the analyzed network traffic with the use of a mixed statistical model (hybrid) ARIMA-GARCH is presented. The introductory actions consisted in normalization of elements of the analyzed time series by means of the Box-Cox transformation. To determine, though, if the analyzed time series were characterized by heteroscedasticity, they were subjected to the White's test. For comparison, there were also tested with the use of differing statistical approaches (described by mean or conditional variance), realized by individual models of ARIMA and GARCH. The choice of optimal models' parameters was performed as a compromise between the coherence of the model and the size of estimation error. To detect attacks (anomalies) in the network traffic, there were used relations between the proper estimated model of the network traffic, and its real parameters. The presented experimental results confirmed fitness and efficiency of the proposed solutions.

Keywords: Time series analysis · Network traffic prediction · Network attacks detection · Hybrid ARIMA-GARCH model

1 Introduction

Dynamic development of threats and incidents violating safety of systems, computer networks, and users of services proposed by modern informational technologies is currently one of the most essential social and civilizational problems. The scope, scale and dynamics of this problem concern not only individual users and small businesses, but also big international corporations, institutions and government agencies.

Until recently, the IT security was ensured by antivirus programmes, firewalls, technical systems, appropriate policies or user trainings. Currently, more and more often, these tools are insufficient. Cybercriminals are always one step ahead. Without ongoing monitoring, companies and institutions will not avoid attacks, and will not protect fully their network resources. Even simple network traffic monitoring may abstract anomalies (possible attacks) and will allow the network administrators to take proper remedial actions. The more advance the monitoring solution is, the greater is

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_1

also the knowledge about the danger and the more accurate protection measure of the whole IT infrastructure in the given organization.

The basic advantage of methods based on anomaly detection is their ability to recognize unknown types of attacks. It is because they do not work on the grounds of knowledge about how a particular attack looks like, but they notice what does not resemble the norms of the network traffic. Therefore, the IDS/IPS systems (Intrusion Detection Systems/Intrusion Prevention Systems) are more efficient than systems utilizing signatures in case of detecting unknown, new types of attacks [1, 2].

Anomaly detection methods have been a field of many studies and review articles [3, 4]. In works describing those methods, the authors used techniques based on machine learning, neural networks or expert systems [5, 6]. Currently, the most extensively developed methods of anomaly detection are those based on statistical modelling that describes the analyzed network traffic. Here, the most often used models are: the autoregressive ARMA or ARIMA, and models with conditional heteroscedasticity, i.e. ARCH and GARCH, which allow for estimation of normal network traffic profiles [7, 8].

The article presented by us uses statistical estimation based on a hybrid model ARIMA-GARCH for particular profiles of the analyzed network traffic behavior. The process of anomaly detection (a network attack) is realized on the grounds of a comparison between normal behavior parameters (predicated on the basis of the examined statistical models) and parameters of the real network traffic.

This paper is organized as follows: after the introduction, in Sect. 2, we present the overview of networks attacks. In Sect. 3, statistical models for network traffic prediction are described in detail. Then, in Sect. 4, the methodology of network attacks detection based on hybrid ARIMA-GARCH model is shown. Experimental results and conclusion are given thereafter.

2 Overview of Networks Attacks

Effective protection of the teleinformatic infrastructure is a major challenge for all entities providing their services in the cyberspace. Standard dangers are viruses, malware and hacking the systems. Lately, however, increasingly more common attacks (due to low costs, facility of realization and high efficiency of impact) are based on blocking access to resources and network services, called DoS/DDoS (Denial of Service/Distributed Denial of Service).

In big simplification, there can be distinguished two main groups of such DoS/DDoS attacks, i.e. attacks of third and fourth layer of OSI reference model (typically network actions), and application layer attacks (most often connected to Web services). In practice, the intruders skillfully connect the mentioned types of attacks, assuming that the more destructive techniques the attack contains, the more effective it is [9].

In general, the DoS attack family, as far as their activity is concerned, can be divided into three kinds. The first type is attacks that are based on implementation of a TCP/IP stack (Transmission Control Protocol/Internet Protocol), which use weaknesses in TCP/IP protocols in given operational system. This group includes: Ping of Death (also known as Long ICMP - Internet Control Message Protocol, which distorts an

ICMP Echo Request packet), Teardrop (it concerns fragmentation of the IP protocols packets and half offset field), SMBnuke and WINnuke. The second type, on the other hand, contains attacks that are based on weaknesses of the TCP/IP stacks' standards. For instance, SYN Flood, Naptha and Land are most often met and dangerous classical attacks which drain the system's resources. Third kind, however, consists of activities such as brute force, which generate big traffic in the network, thereby they exhaust the IT infrastructure's available bandwidth. The most popular examples are Smurf, UDP Flood, Fraggle, Pingflood and Jolt [10].

However, the most common DDoS attacks are: Torinoo (UDP (User Datagram Protocol) flood), Mstream (TCP ACK Flood + IP spoofing), Shaft (UDP + TCP SYN + ICMP flood and conducting statistics), Stacheldraht (UDP + TCP SYN, ACK, NULL + ICMP flood, TFN (UDP + ICMP Echo + TCP SYN flood and Smurf attack), TFN2 K (UDP + TCP + ICMP flood), Smurf attack, Targa3 and IP spoofing. For building the so called DDoS network there are also used such elements as: exploits (they are used for obtaining the administrator's rights), rootkits (they are used to disguise the hacking of the attacked system), ports' scanners (utilized for searching new victims), sniffers (realizing data eavesdropping), autorooters (Trojan horses which introduce automation to the DDoS network construction), and daemons (background processes, operating without interaction with the user) [9].

Undoubtedly, the attacks of the application layer, often called AppDoS (Application Denial of Service), are much more dangerous and difficult to counter. They mostly aim at Web services. One of the reasons for their effectiveness is that it is difficult to distinguish normal increase in the service interest from the attack itself.

The approaches most often used for detection of such attacks are based on methods utilizing statistical anomaly detection on the grounds of estimated specific profiles of the network traffic. The profiles are usually characterized by average size of network traffic components, i.e. the number of IP packages, average number of newly established connections within a time unit, the ratio of packages of particular network protocols, etc. One can also observe and use some statistical dependences resulting from the time of the day or week, as well as keep statistics for particular network protocols. NIDS systems (Network-based Intrusion Detection System) based on these methods can learn a typical network's profile – this process lasts from few days to few weeks – and later compare the current activity in the network with created, typical profile. This comparison will constitute grounds for verification if there is anything unusual happening (a network attack) [1, 11].

3 Statistical Model for Network Traffic Prediction

Mostly, current research connected to the statistical analysis of time series (predication in particular) concerns processes that are characterized by lack of or weak connection between the variables which are separated by some time period. However, in many practical applications, there is a need for modelling processes of which the autocorrelation function slowly decreases, and the relation between distant observations, even though it is small, is still essential.

Long term dependences are visible in existence of autocorrelation of elements creating the given time series. In most cases, it is the high order correlation. This means, that in the tested series, there is a dependence between observations, even those much distant in time. Such phenomenon is known as long memory, and was discovered by a British hydrologist, Hurst.

An interesting approach towards description of attributes of long memory time series was the use of autoregressive solution with moving averaging in the differentiation process. As a result, the ARIMA model (Autoregressive Integrated Moving Average) [12] emerged, which is a generalization of ARMA models for non-stationary processes.

A different approach to description of time series took into account the process's conditional variance dependence on its previous values by means of ARCH model (Autoregressive Conditional Heteroskedastic Model), introduced by Engel [13]. This model's generalization was the GARCH model (Generalized Autoregressive Conditional Heteroscedasticity) [14], whose autocorrelation function of the model's squared residuals decreases in hyperbolic manner. Such behavior of autocorrelation function enables to call GARCH a model of long memory in the context of autocorrelation function of the model's squared residuals [12, 15].

3.1 The ARIMA Model

The ARMA model is only suitable for the analysis of stationary series [16]. When an ARMA series is non-stationary, it needs to be differenced at least once to produce a stationary series. The result of this operation is Autoregressive Integrated Moving Average (ARIMA) model for time series y_t of order (p, d, q) which can be expressed as [14]

$$\Phi(L)(1-L)^d y_t = \Theta(L)\epsilon_t, \qquad t = 1, 2, \ldots T, \tag{1}$$

where y_t is the time series, $\epsilon_t \sim (0, \sigma^2)$ is the white noise process with zero mean and variance σ^2, $\Phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p$ is the autoregressive polynomial and $\Theta(L) = 1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q$ is the moving average polynomial, L is the backward shift operator and $(1-L)^d$ is the fractional differencing operator given by the following binomial expansion:

$$(1-L)^d = \sum\nolimits_{k=0}^{\infty} \binom{d}{k} (-1)^k L^k \tag{2}$$

and

$$\binom{d}{k}(-1)^k = \frac{\Gamma(d+1)(-1)^k}{\Gamma(d-k+1)\Gamma(k+1)} = \frac{\Gamma(-d+k)}{\Gamma(-d)\Gamma(k+1)}. \tag{3}$$

$\Gamma(*)$ denotes the gamma function and d is the number of differences required to give stationary series and $(1-L)^d$ is the d^{th} power of the differencing operator. If no differencing is done $(d=0)$, the models are usually referred to as ARMA (p,q) [18].

Forecasting ARIMA processes is usually carried out by using an infinite autoregressive representation of (1), written as $\Pi(L)y_t = \epsilon_t$, or

$$y_t = \sum\nolimits_{i=1}^{\infty} \pi_i y_{t-i} + \epsilon_t, \tag{4}$$

where $\Pi(L) = 1 - \pi_1 L - \pi_2 L^2 - \ldots = \Phi(L)(1-L)^d \Theta(L)^{-1}$.

In terms of practical implementation, this form needs truncation after k lags, but there is no obvious way of doing it. This truncation problem will also be related to the forecast horizon considered in predictions (see [18]). From (4) it is clear that the forecasting rule will pick up the influence of distant lags, thus capturing their persistent influence. However, if a shift in the process occurs, this means that pre-shift lags will also have some weight on the prediction, which may cause some biases for post-shift horizons [12].

3.2 The GARCH Model

The extension of ARCH model is the Generalized ARCH model [14, 19], introduced by Bollerslev and Taylor. The GARCH (r,s) model is given by (5) along with the volatility equation

$$h_t = \alpha_0 + \sum\nolimits_{i=1}^{r} \alpha_i \epsilon_{t-i}^2 + \sum\nolimits_{j=1}^{s} \beta_j \epsilon_{t-j}^2, \tag{5}$$

where the model's parameters adept values $r > 0$, $s > 0$, $\alpha_0 > 0$ and $\alpha_i > 0$ for $i = 1, 2, \ldots, r$ and $\beta_j > 0$ for $j = 1, 2, \ldots, s$. If we assume that $\alpha(L)$ and $\beta(L)$ are lag operators, such as $\alpha(L) = \alpha_1 L + \alpha_2 L^2 + \cdots + \alpha_r L^p$ and $\beta(L) = \beta_1 L + \beta_2 L^2 + \cdots + \beta_s L^q$, then the equation of variability (5) adopts the form

$$h_t = \alpha_0 + \alpha(L)\epsilon_t^2 + \beta(L)h_t. \tag{6}$$

For $s = 0$, the model reduces to an ARCH (r) and for $r = s = 0$, ϵ_t is simply a white noise process [19].

In practice, a commonly used specification is the GARCH model (1,1). It allows for a decent description of variability of the examined time series. Its advantage over ARCH lies in the fact that there is a small number of used parameters and, therefore, there is optimization of the speed and computational complexity [15].

The forecasts of the GARCH model are obtained recursively [14, 15]. The general j-step ahead forecast for h_{k+j}^2, at the forecast origin k, is

$$h_k^2(j) = \alpha_0 + (\alpha_1 + \beta_1) h_t^2(j-1), \quad \text{for} \quad j > 1. \tag{7}$$

Repeating the substitutions for $h_k^2(j-1)$ until the j-step forecast can be written as a function of $h_k^2(1)$ gives the explicit expression for the j-step ahead forecast

$$h_k^2(j) = \frac{\left[1-(\alpha_1+\beta_1)^{j-1}\right]}{1-\alpha_1-\beta_1} + (\alpha_1+\beta_1)^{j-1} h_k^2(1). \tag{8}$$

In practice, for prediction purposes, a commonly used specification is the GARCH (1,1) model. It allows for decent description of the tested time series variability. Whereas its advantage over the ARCH model consists in small application possibilities of parameters, similarly its speed optimization and computational complexity level [19].

3.3 The Hybrid ARIMA-GARCH Model

Creation of the hybrid ARIMA-GARCH model is performed in two stages. In the first stage, best fitted ARIMA model is sought in order to model the linear factor of the analyzed time series. In the second stage, GARCH is used to describe the non-linear residual factors (residuals of the model) [20]. Then, the analytic form of the hybrid ARIMA-GARCH model (p, d, q, r, s) can be presented as:

$$\Phi(L)(1-L)^d y_t = \Theta(L)\epsilon_t, \qquad \epsilon_t \sim \left(0, h_t^2\right) \qquad t = 1, 2, \ldots T, \tag{9}$$

$$h_t = \alpha_0 + \alpha(L)\epsilon_t^2 + \beta(L)h_t. \tag{10}$$

The prediction process of the future values of the analyzed time series in the ARIMA-GARCH model was described in detail by Liu Heping and Shi Jing in their work [20].

4 Methodology of Network Attacks Detection

The process of attack (anomalies) detection was based on parameters comparison of the estimated normal behavior by means of described models and variability parameters of the real network traffic. In the initial stage of analysis, there was performed normalization of the tested time series with the use of Box-Cox transformation. The choice of optimal parameter values of the analyzed statistical models was realized as a compromise between the coherence of the given model and the size of its estimation error. The White's test, however, was used for detection of heteroscedasticity in the statistical modelling process of the tested time series.

4.1 The Box-Cox Transformation of Analyzed Time Series

Normalization of the time series elements was performed by means of exponential transformation

$$y = \left\{ \begin{array}{ll} \left(x^{\lambda} - 1\right), & \lambda \neq 0 \\ \ln x, & \lambda = 0 \end{array} \right\}, \tag{11}$$

where x is a time series element, and λ is the exponential transformation parameter [21].

Parameter λ can be estimated based on the given time series elements $x_1, x_2, \ldots, x_n$ with the use of maximum likelihood method, and then it narrows to finding the maximum of the function

$$L(\lambda; x_1, x_2, \ldots x_n) = -\frac{n}{2}\ln\left[\sum\nolimits_{i=1}^{n} y_i^2 - \frac{1}{n}\left(\sum\nolimits_{i=1}^{n} y_i\right)^2\right] + (\lambda - 1)\sum\nolimits_{i=1}^{n} \ln x_i \tag{12}$$

or to solution of the equation

$$\frac{n\sum_{i=1}^{n} u_i y_i - \left(\sum_{i=1}^{n} u_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2} - \frac{1}{n}\lambda\sum\nolimits_{i=1}^{n} \ln x_i = 1, \tag{13}$$

where $u = x^{\lambda} \ln x$.

4.2 The White's Test for Heteroscedasticity Occurrence

Inference based on the statistical model in which we will omit the problem of heteroscedasticity may be incorrect. Thus, an important element of creation of a correct model is the examination whether the random factor is heteroscedastic. Most tests detecting heteroscedasticity are based on the fact that the estimator of the least squares method is consistent, even if heteroscedasticity is present. Therefore, the model's residuals obtained by means of the least squares method will behave very similarly to real residuals, even with heteroscedasticity. Bearing in mind this attribute, the test is constructed on the basis of the received residuals from regression.

The White's test is a general tool detecting presence of heteroscedasticity, by which we verify the following hypotheses [14, 15]:

- H_0: the random factor is homoscedastic,
- H_1: the random factor is heteroscedastic (contradiction of H_0).

The test consists of three phases. In the first one, we estimate the initial regression model $Z = \beta y + \epsilon$ and we remember the vector of residuals$\hat{\epsilon}$. Then, we perform linear regression of the variable $\hat{\epsilon}^2$ on the constant, squares of explanatory variables, and all the mixed products of explanatory variables, and as a result we obtain the determination coefficient R^2. By fulfilling the assumption H_0 statistics nR^2 on distribution χ^2. Next, we calculate the size of statistics nR^2, and on such basis we verify the hypothesis H_0.

Intuitively, the idea of the test is simple. If the model is correct, and heteroscedasticity is not present, the squares of residuals should not explain much. Thus, if the testing statistics is small, we do not have grounds for stating that heteroscedasticity is present in the model.

4.3 The Choice of the Model and Its Parameters' Estimation

The methods often used for parameters' estimation in autoregressive models ARIMA and GARCH are: the maximum likelihood method (MLE) and the quasi-maximum likelihood method (QMLE). It results from the fact that parameters' estimation by means of both methods is relatively simple, quick and effective. However, the main computational problem of the two methods is finding a solution to the equation

$$\frac{\partial ln(L_{\Omega}(\varrho))}{\partial \theta} = 0, \tag{14}$$

where θ is the estimated set of parameters, $L_{\Omega}(\varrho)$ is the likelihood function, and Ω is the number of observations. Most often, in general case, analytic solution to the Eq. (14) is impossible, thus, numerical estimation is used.

The basic problem appearing while using the maximum likelihood method is necessity to determine the whole model, and, in consequence, sensitivity of the obtained estimator to possible errors in the AR and MA polynomials, which answer for the process's dynamics.

There is no universal criterion for choosing the form of the model. Usually, the more complex the model is, the highest is its likelihood function's value, and then matching the model to the data is more optimal. However, estimation of greater number of parameters is usually burdened with bigger errors. Therefore, a form of compromise is sought between the number of parameters occurring in the model, and the value of likelihood function. The choice of sparse representation of the model is most often performed on the basis of information criteria, such as the Akaike's (AIC), Schwarz's (SIC), or Hannan-Quinn's (HQC). Then, out of different forms of the model, the one that is chosen has the lowest value of information criterion [16, 22].

In this work, for the parameters' estimation and for selecting the form of the model we used the maximum likelihood method. The choice was based on the method's relative simplicity and computational efficacy. For ARIMA model we used the automatic selection algorithm of the row of the model based on information criteria (see Hyndman and Khandakar [23]). For the GARCH model estimation, on the other hand, we used methodology described in work [17].

5 Experimental Result

To check usefulness of the proposed anomaly detection method we used testbed based on LAN network with SNORT [24] anomaly detection preprocessor, where anomaly detection algorithms were implemented. We used similar network architecture which we proposed in [25]. Network traffic features are collected by SNORT IDS (see Table 1). We used C1–C26 LAN network traffic features in order to evaluate performance of the proposed anomaly detection method.

We simulated different network attacks/anomalies in a controlled network environment. Attacks were performed by means of Kali Linux [26] toolset implemented in this Linux distribution. Attacks/anomalies belong to different groups, such as: DoS,

Table 1. Description of C1–C26 network traffic features

Feature	Traffic feature description	Feature	Traffic feature description
C1	Number of TCP packets	C14	Out TCP packets (port 80)
C2	In TCP packets	C15	In TCP packets (port 80)
C3	Out TCP packets	C16	Out UDP datagrams (port 53)
C4	Number of TCP packets in LAN	C17	In UDP datagrams (port 53)
C5	Number of UDP datagrams	C18	Out IP traffic [kB/s]
C6	In UDP datagrams	C19	In IP traffic [kB/s]
C7	Out UDP datagrams	C20	Out TCP traffic (port 80) [kB/s]
C8	Number of UDP datagrams in LAN	C21	In TCP traffic (port 80) [kB/s]
C9	Number of ICMP packets	C22	Out UDP traffic [kB/s]
C10	Out ICMP packets	C23	In UDP traffic [kB/s]
C11	In ICMP packets	C24	Out UDP traffic (port 53) [kB/s]
C12	Number of ICMP packets in LAN	C25	In UDP traffic (port 53) [kB/s]
C13	Number of TCP packets with SYN and ACK flags	C26	TCP traffic (port 4444)

APPDDoS - application specific DDoS, DoS, different methods of port scanning, DDoS, packet fragmentation, Syn Flooding, reverse shell, spoofing, and others.

For statistical algorithms network traffic features C1–C26 are converted to subsequent time series representing these features. After time series preprocessing (for e.g. time series normalization see Sect. 4.1), the SNORT preprocessor calculates models parameters for traffic C1–C26 features (we control our network and we assume that there is no anomalies/attacks during this period of time). We achieve for every model forecasting prediction intervals profiles (for example, 30 samples prediction intervals [25]). Prediction intervals profiles are used to detect possible anomalies/attacks during normal work of the proposed network preprocessor. If network traffic features values C1–C26 exceed the range specified by calculated prediction intervals, we assume a possible anomaly/attack.

Results of the comparison between ARIMA, GARCH and ARIMA-GARCH statistical models for network anomaly detection are presented in Table 2 (DR[%] values) and Table 3 (FP[%] values). For a given 26-traffic-features set and simulated attacks/anomalies, the best results of DR and FP were achieved for ARIMA-GARCH model. Worse results were achieved for GARCH and ARIMA models. GARCH model gives us slightly better results than ARIMA model. Hybrid ARIMA-GARCH model uses best features of both models (see explanation in Sect. 3.3), which is why we were able to achieve better results than in the case of ARIMA and GARCH. FP were under 10%, while DR achieved 90%. Some low values in Tables 2 and 3 require more

Table 2. Results of detection rate DR[%] for C1–C26 network traffic feature

Feature	ARIMA	ARIMA-GARCH	GARCH	Feature	ARIMA	ARIMA-GARCH	GARCH
C1	3.10	4.20	3.26	C14	8.16	9.12	8.84
C2	8.12	9.12	8.84	C15	8.16	9.12	8.84
C3	8.12	9.12	8.84	C16	0.00	0.00	0.00
C4	8.12	9.12	8.84	C17	3.45	4.56	2.45
C5	8.12	9.12	8.84	C18	8.16	9.12	8.84
C6	0.00	0.00	0.00	C19	8.16	9.12	8.84
C7	0.00	0.00	0.00	C20	3.22	4.56	2.45
C8	28.64	30.20	26.54	C21	8.48	9.24	8.84
C9	85.46	90.24	83.42	C22	0.00	0.00	0.00
C10	85.46	90.34	82.22	C23	0.00	0.00	0.00
C11	0.00	0.00	0.00	C24	0.00	0.00	0.00
C12	75.26	78.84	73.12	C25	0.00	0.00	0.00
C13	8.16	9.12	6.34	C26	74.42	76.00	72.21

Table 3. Results of false positive rate FP[%] for C1–C26 network traffic feature

Feature	ARIMA	ARIMA-GARCH	GARCH	Feature	ARIMA	ARIMA-GARCH	GARCH
C1	7.44	5.24	6.22	C14	7.54	5.25	6.24
C2	7.42	5.42	6.15	C15	8.48	5.43	7.22
C3	7.44	5.22	6.15	C16	4.24	3.24	3.75
C4	7.62	5.12	6.17	C17	4.14	3.52	4.29
C5	6.14	4.42	5.32	C18	7.64	5.12	6.35
C6	6.16	4.26	4.56	C19	7.26	5.23	6.62
C7	7.32	6.84	6.92	C20	8.24	6.72	7.12
C8	8.76	6.42	7.21	C21	8.26	5.32	7.14
C9	9.14	7.74	8.14	C22	5.32	4.21	5.14
C10	4.24	3.82	4.46	C23	7.46	5.45	6.26
C11	5.22	4.27	5.44	C24	0.00	0.00	0.00
C12	3.22	2.24	3.25	C25	2.21	1.05	1.82
C13	8.12	6.14	7.63	C26	2.21	1.45	1.82

explanation. For example, C1, C6 and C7 have low values because simulated anomalies/attacks do not have an impact on these features.

6 Conclusion

Ensuring an appropriate level of safety for resources and IT infrastructure systems is currently an extensively studied and developed issue. It is obvious that wired LAN/WAN networks, and radio Wi-Fi/WSN networks, due to their nature, are exposed to a significant number of hazards, coming from both: the outside and inside of their infrastructure. Therefore, those networks require providing integrity and confidentiality of transmission, but also protection of the data sent with their use.

A constantly growing number of new attacks, their global scope and level of complexity enforce dynamic development of the network protection systems. The most often implemented mechanisms, serving to provide this safety, are methods of detection and classification of abnormal behaviors (usually being a consequence of an attack) reflected in the analyzed network traffic.

The strength of such an approach is protection from attacks not known so far, which were developed deliberately (targeted attacks) for realization of activities on particular network infrastructure resources, or simply constituting the, so called, zero-day exploits. The attacks (anomalies) detection systems can have special input in such environments. The role of such systems is then detection (for the purpose of automatic reaction) of not typical behaviors in the network traffic which are symptoms of unauthorized actions (attacks), directed against the protected IT infrastructure resources.

This work presents a detection method of attacks onto (anomalies in) the network traffic parameters, with the use of a mixed statistical model (hybrid) ARIMA-GARCH. To detect anomalies in the network traffic there was conducted research on the differences between the real traffic and the estimated model of that traffic for the analyzed network parameters. The choice of the sparingly parameterized form of the model was made on the basis of a compromise between the sparse representation and the size of estimation error. In the proposed method, statistical relations between the estimated traffic model and its real variance were used to detect abnormal behavior, which is possibly an aftermath of a network traffic attack. The obtained experimental results confirm efficiency of the presented method, and accuracy of the choice of statistical models for the analyzed parameters of the protected network.

References

1. Axelsson, S.: Intrusion detection systems: A survey and taxonomy. Technical report 99-15, Department of Computer Engineering (2000)
2. Jackson, K.: Intrusion Detection Systems (IDS), Product Survey. Los Alamos National Library, LA-UR-99-3883 (1999)
3. Hajji, H.: Statistical analysis of network traffic for adaptive faults detection. IEEE Trans. Neural Netw. **16**(5), 1053–1063 (2005)
4. Kiedrowski, P.: Toward more efficient and more secure last mile smart metering and smart lighting communication systems with the use of PLC/RF hybrid technology. Int. J. Distrib. Sens. Netw. **2015**, 1–9 (2015). Article ID 675926. http://dx.doi.org/10.1155/2015/675926
5. Barford, P., Kline, J., Plonka, D., Ron, A.: A signal analysis of network traffic anomalies. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, pp. 71–82. ACM (2002)
6. Chondola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 1–72 (2009)
7. Yaacob, A., Tan, I., Chien, S., Tan, H.: ARIMA based network anomaly detection. In: Second International Conference on Communication Software and Networks, pp. 205–209. IEEE (2010)

8. Rodriguez, A., Mozos, M.: Improving network security through traffic log anomaly detection using time series analysis. In: Computational Intelligence in Security for Information Systems, pp. 125–133 (2010)
9. Liang, H., Xiaoming, B.: Research of DDoS attack mechanism and its defense frame. In: 3rd International Conference on Computer Research and Development, pp. 440–442 (2011)
10. Mirkovic, V.J., Prier, G., Reiher, P.: Attacking DDoS at the source. In: Proceedings of ICNP, pp. 312–321 (2002)
11. Thottan, M., Ji, C.: Anomaly detection in IP networks. IEEE Trans. Sig. Process. Spec. Issue Sig. Process. Netw. **51**(8), 2191–2204 (2003)
12. Granger, C.W.J., Joyeux, R.: An introduction to long-memory time series models and fractional differencing. J. Time Ser. Anal. **1**, 15–29 (1980)
13. Engle, R.: Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation. Econometrica **50**, 987–1008 (1982)
14. Bollerslev, T.: Generalized autoregressive conditional heteroskedasticity. J. Econometrics **31**, 307–327 (1986)
15. Tayefi, M., Ramanathan, T.V.: An overview of FIGARCH and related time series models. Austrian J. Stat. **41**(3), 175–196 (2012)
16. Box, G., Jenkins, G., Reinsel, G.: Time Series Analysis. Holden-day, San Francisco (1970)
17. Baillie, R., Bollerslev, T., Mikkelsen, H.: Fractionally integrated generalized autoregressive conditional heteroskedasticity. J. Econometrics **74**, 3–30 (1996)
18. Hosking, J.: Fractional differencing. Biometrika **68**, 165–176 (1981)
19. Andersen, T.G., Bollerslev, T.: ARCH and GARCH models. In: Kotz, S., Read, C.B. Banks, D.L. (eds.) Encyclopaedia of Statistical Sciences, vol. 2. John Wiley and Sons (1998)
20. Liu, H., Shi, J.: Applying ARMA-GARCH approaches to forecasting short-term electricity prices. Energy Economics **37**, 152–166 (2013)
21. Box, G.E.P., Cox, D.R.: An analysis of transformations. J. Roy. Stat. Soc. B **26**, 211–252 (1964)
22. Brockwell, P., Davis, R.: Introduction to time series and forecasting. Springer Verlag (2002)
23. Hyndman, R.J., Khandakar, Y.: Automatic time series forecasting: the forecast package for R. J. Stat. Softw. **27**(3), 1–22 (2008)
24. SNORT - Intrusion Detection System. https://www.snort.org
25. Andrysiak, T., Saganowski, Ł., Choras, M., Kozik, R.: Network traffic prediction and anomaly detection based on ARFIMA model. In: de la Puerta, J.G. et al., International Joint Conference SOCO’14-CISIS’14-ICEUTE’14 Advances in Intelligent Systems and Computing, vol. 299, pp. 545–554 (2014)
26. Kali Linux. https://www.kali.org

Towards Mixed-Mode Risk Management – A Concept

Andrzej Bialas(✉) and Barbara Flisiuk

Institute of Innovative Technologies EMAG,
Leopolda 31, 40-189 Katowice, Poland
{andrzej.bialas,barbara.flisiuk}@ibemag.pl

Abstract. The paper concerns the risk management issue. Different approaches – static and real-time (dynamic) are reviewed, as well as their advantages and gaps. A broadly used static risk assessment/management (SRA/M) process is comprehensive, complex, invoked periodically, but it omits fluctuating risk factors and has limited ability to adapt itself to the changing risk picture. The paper proposes a mixed-mode approach. The SRA/M process is transformed towards the adaptive risk management (ARM) process. This adaptation is based on real-time risk management (RTRM) results gathered during a period between static risk assessments. All risk management processes are expressed in a pseudo-code. The method is exemplified by a simple but representative case study.

Keywords: Risk management · Real-time risk management · Adaptive risk management

1 Introduction

The paper concerns an event-based advanced risk management methodology which embraces static and dynamic aspects of the risk nature in security.

Risk management is a continuous process. It includes the identification, analysis, and assessment of potential hazards in a system or hazards related to a certain activity. Once the risk picture is recognized, there are some risk control measures proposed to eliminate or reduce potential harms to people, environment, or other assets. In addition, the risk management process embraces risk monitoring and communication. ISO 31000 [1] is the basic risk management standard. Examples of the most recognized risk management methods and techniques are included in ISO/IEC 31010 [2]. This issue was discussed in [3–5].

The methods described in the standards have static character, are performed periodically and are able to analyze steady state and low frequency risk factors. The main gap in this approach is related to the fact that fluctuating threats or hazards, occurring between the performed analyses, are omitted, though they can be dangerous as well.

The objective of the paper is to analyze whether fluctuating, dynamic phenomena may be taken into consideration during the risk management process and to present a new mixed-mode approach.

Section 2 includes a short review of static and dynamic risk management approaches. Section 3 presents the static approach and how to integrate real-time risk

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_2

management in this approach to obtain the mixed-mode approach. Section 4 discusses a short example of the mixed-mode method, while the final section concludes research results and discusses further steps.

2 Risk Management – Static, Iterative and Real-Time Approaches

Today's risk analysis methods (including those described in [2–4, 6]) are based mainly on a static risk management approach. This approach is typically applied in strategic risk management where the risk is assessed based on static factors existing over relatively long periods of time. The analyses are made periodically with respect to the current situation and historical data. However, risk levels that are determined might not reflect adequately the changes occurring in the business environment, as fluctuating risk factors are not taken into consideration.

These drawbacks can be eliminated by means of the iterative risk management approach (IRM), also called adaptive risk management approach (ARM). For example, the paper [7] uses this approach to mitigate the uncertainty problem during a sample risk analysis for the technical component used in the oil and gas industry. These analyses are conducted periodically too, still, each successive analysis is made in an improved manner (with respect to the knowledge acquired over time). The idea is to use the experiences learnt from the operation of security measures that had been selected and implemented based on the previous analysis. Though this approach is definitely a flexible one, it is still related to some disadvantages. For example, it is difficult to identify suitable risk thresholds when multiple risks are acting together, and when different scenarios and threshold assumptions have to be considered. Whenever the methods and thresholds change, it is difficult to compare the results.

As fluctuating risk factors and the related increased uncertainty are disregarded, currently used risk management methods of these two approaches are often ineffective. Thus it is necessary to employ risk management methods that would assess and reduce risk in real time. Static and iterative methods should be supported by methods that would adapt dynamically to the fast changing picture of threats and changes in the protected objects (their assets and vulnerabilities). The newly developed solutions should provide on-line communication, statistical data about incidents to support risk management, data from security analyzers or on-line security monitoring systems.

Cyber security needs on-line monitoring of the existing risks due to fast changes of fluctuating risk factors. It deals with both previously identified and emerging threats and vulnerabilities. New technologies, such as cloud computing, mobile computing, virtualization, Internet of Things, etc., generate new, specific threats and vulnerabilities. This was motivation to elaborate real-time risk management (RTRM) approaches which are typical of operational risk management. According to [8], RTRM systems should comprise the following elements:

- instantaneous knowledge – changes in assets, threats, vulnerabilities, risk categories and levels should be visible instantly;

- comprehensive visibility – to have a clear picture of assets and the related vulnerabilities and potential impacts, one must have consistent data, visibility, and alerts;
- constant controls assessment and adjustment – security measures should be assessed periodically and persistently in order to address new or changing risks.

The objective of the real-time risk management approach is to view the entire IT infrastructure in this respect and ensure a deep technical insight into each technology of this infrastructure. This approach makes it easier to monitor security events in real time, to find their correlations, to interpret, alarm and visualize the current security picture. According to [8], advanced real-time risk management systems have to be supported by uninterrupted threat monitoring, knowledge correlation and alerting engines, and, as far as possible, automatic response.

The role of the Security Information and Event Management (SIEM) has been growing recently [9, 10]. SIEM systems are designed to aggregate, analyze and present large volumes of security-related data acquired from the network and security devices with a view to detecting threats, incidents, vulnerabilities, frauds, including proper reaction and warning against them.

Cyber security cannot be properly managed when its status is not measured. Measures can be based on indicators coming from different sources:

- SIEM and data analytics tools which provide valuable information on actual or potential compromise on the network,
- threat intelligence services, compliance management, vulnerability management, penetration testing and audits are helpful to identify data losses and provide valuable information on actual or impending attacks.

The holistic approach is recommended – using all this information [11] the cyber security status can be identified and presented to decision makers. A very important issue is to maintain and share security-related information [12, 13].

Real-time risk management is used mainly for cyber security risks. There are no integrated, holistic approaches that would embrace the static-, adaptive- and real-time risk management. It can be strong motivation for researchers to work out methods and tools in this field. The paper proposes a concept of such a method.

There are many event-based risk management techniques applied in the security domain. Similar techniques exist in the safety domain, along with system theory-based techniques to better tackle issues in this domain, such as complex software-intensive systems, complex human-machine interactions and systems-of-systems [14].

3 Mixed-Mode Risk Management Approach – A Concept

A mixed-mode risk management concept is based on the static risk management loop supplemented by real-time risk management facilities. The methodology should comply with ISO 31000 [1]. There are a lot of static risk analysis methods [2]. The elaborated mixed-mode risk management framework will be based on a commonly known and relatively simple method: consequence-probability matrix. Other, more complex methods can be considered in the future. Both process- and asset-oriented approaches are possible but the paper is focused on the asset-oriented one.

3.1 Static Risk Management (SRM) Process

The static risk assessment is performed periodically, e.g. yearly (Organizational Security Policy specifies "when"). Let us assume that when this time occurs the SRA_event is triggered, initiating the main loop of the SRM process – presented below in a pseudo-code. For any quadruple <asset, threat, vulnerability, existing countermeasures> the likelihood and consequences of the hazardous event caused by a threat are assessed and then the risk level is calculated using these two parameters (usually as a product of them).

```
IF SRA_event THEN
 FOR each identified elementary asset
  FOR each threat relevant to elementary asset
   FOR each vulnerability relevant to threat
    COMMENT start existing risk assessment (“as-is”)
    BeforeCountermeas :=
          IdentifyExistingCountermeasures()
    BeforeLikelihoodLevel := AssessLikelihood()
    BeforeConsequencesLevel := AssessConsequences()
    BeforeRiskLevel :=
          BeforeLikelihoodLevel*BeforeConsequencesLevel
    COMMENT initializing variables for main
    risk management loop
    AfterCountermeas := BeforeCountermeas
    AfterLikelihoodLevel := BeforeLikelihoodLevel
    AfterConsequencesLevel := BeforeConsequencesLevel
    AfterRiskLevel := BeforeRiskLevel
    COMMENT main risk management loop
     WHILE AfterRiskLevel > RiskAcceptanceLevel
      AfterCountermeas :=
           SelectBetterCountermeasures()
      AfterLikelihoodLevel := AssessLikelihood()
      AfterConsequencesLevel := AssessConsequences()
      AfterRiskLevel :=
           AfterLikelihoodLevel*AfterConsequencesLevel
     ENDWHILE
   ENDFOR (vulnerabilities)
  ENDFOR (threats)
 ENDFOR (assets)
ENDIF (SRA_event)
```

As a result of that, the risk pictures before and after the mitigation are identified. Please note that the heuristic, human activities are expressed as "`functions()`", and marked *italic*.

Risk management embraces risk analysis, risk assessment, and the selection of countermeasures, when the current risk level (`AfterRiskLevel`) exceeds the `RiskAcceptanceLevel`. The method assumes that during the preliminary analysis (`BeforeRiskLevel`) certain countermeasures exist and mitigate the risk to a certain extent, usually insufficiently. These measures ought to be revised. They can be accepted or replaced by their more effective combinations, and then the risk should be reassessed. Countermeasure is understood here as a coherent, diversified set of elementary security measures.

Summing up, for any triple <asset, threat, vulnerability>, called risk scenario, the risk before and after reduction is assessed and the existing and applied countermeasures are specified as the static risk record:

`srr=(asset, threat, vulnerability, BeforeCountermeas, BeforeLikelihoodLevel, BeforeConsequencesLevel, BeforeRiskLevel, AfterCountermeas, AfterLikelihoodLevel, AfterConsequencesLevel, AfterRiskLevel).`

The static risk can be considered as a sum of these records for all scenarios:

$$SR = \bigcup\nolimits_{all\,scenarios}\{srr\} \tag{1}$$

The static risk (SR) assessment creates a general picture of the risk situation in a certain moment of the system life cycle. These moments can be shown as points on a discrete time scale: $SR(T_0)$, $SR(T_1)$, $\ldots SR(T_i)$, $SR(T_{i+1})$, $SR(T_{i+2})$, $\ldots SR(T_n)$.

Please note that $SR(T_{i+1})$ is determined using the risk factors identified at the end of $SR(T_i)$ and it represents the current static risk picture, valid until the new analysis result, i.e. $SR(T_{i+2})$ replaces the old one, e.g. after one year. Between any consecutive time points many unpredicted, negative phenomena may occur. The time period between these points is called the SRA period. It is difficult to raise the frequency of static risk assessments to increase the preciseness of assessments because SRA is a time and cost consuming process. RTRM may be helpful to solve this problem.

3.2 Real-Time (Dynamic) Risk Management (RTRM) Process

Real-time risk assessment allows to detect changes of risk relevant factors which influence the overall risk picture and its elements, like: assets, threats, vulnerabilities, countermeasures, likelihood, consequences, etc. RTRM allows to react to these changes as well. This reaction occurs in a relatively short time, allows to revise the current static risk picture and to properly respond to incidents or their symptoms.

Please note that during the previous static risk assessment certain threats or vulnerabilities may be unknown, and some of them may not be identified properly. For this reason the security system (set of the managed countermeasures) is unable to react to them. Similarly, after the previous static risk assessment some organizational changes may occur as well as changes in the business processes, IT infrastructure or business environment. Due to that the countermeasures, selected or updated last time, do not suit the new situation.

The risk assessment and change management shortcomings may cause security incidents, which should be properly managed (reaction, mitigation of damages, lessons learnt, corrective actions within the protection system). Factors, e.g. incident symptoms, non-compliances should be identified immediately and properly managed by the real-time risk management process. The factors influencing a current risk level are diversified, derived from different sources and have different dynamics and nature. The paper proposes means and ways to manage them in a unified way and to identify how the current risk factors modify the previously identified static risk picture.

The real-time risk management process needs input, i.e. information about risk relevant factors. Table 1 presents six sources of information about incidents or their symptoms which can be considered as the input for the real-time risk management process. They should be compared with these specified in the paper [11]. The SEM (Security event management) and SIM (Security information management) systems are distinguished because they have different input. Both can be replaced by SIEM. Sources are activated on request when a behaviour anomaly is detected, when an incident occurs or a certain variable exceeds the assumed level. These sources have different response (identification) time and nature with respect to the incident or its symptom. For each component representing sources, the main outputs are specified along with the activities required in the RTRM process. "Ex ante" concerns incident symptoms, and "ex post" – incident consequences.

Sources of information constitute an intermediate layer of components between the secured object and the RTRM process. The layer is identified and presents, in a unique way, all risk relevant factors whose changes may influence the current overall risk picture. Based on this input, the RTRM process will focus only on the areas of the risk picture which are impacted by these factors. The areas may concern all scenarios (called RT scenarios) related to the given asset or group of assets.

The real-time risk management process runs similarly to SRM, though it is preceded by the preliminary stage of identifying the RT scenarios, which should be reassessed by the RTRM process. This is necessary for SEM, SIM, Penetration tests, and vulnerability scanners, because their output usually points at threats and vulnerabilities only. For this reason, during the preliminary stage all relevant assets for pairs <threat, vulnerability> should be identified. These activities are represented below by the *CompleteRiskScenarios()* function. Generally it is an operation on the database storing the risk scenarios.

One or more completed RT scenarios <asset, threat, vulnerability> are provided as the input for the RTRM process. These scenarios define an area of the overall risk picture which requires reassessment, because the changed risk-related factors may influence the risk level in this area.

Any change in any source component output generates the event `RTRM_event` initiating the RTRM process:

```
IF RTRM_event THEN
 CompleteRiskScenarios()
 COMMENT one or more completed risk scenarios defining
 the possible impacted area as the RTRM input
 FOR each identified elementary asset belonging to
  the RT scenarios subset
  FOR each threat relevant to elementary asset
   FOR each vulnerability relevant to threat
    COMMENT start existing risk assessment
    BeforeRTRMCountermeas :=
          IdentifyExistingRTRMCountermeasures()
    BeforeRTRMLikelihoodLevel := AssessLikelihood()
    BeforeRTRMConsequencesLevel := AssessConsequences()
    BeforeRTRMRiskLevel :=
          BeforeRTRMLikelihoodLevel
          *BeforeRTRMConsequencesLevel
    COMMENT initializing variables for main
    RTRM risk management loop
    AfterRTRMCountermeas :=
           BeforeRTRMCountermeas
    AfterRTRMLikelihoodLevel :=
           BeforeRTRMLikelihoodLevel
    AfterConsequencesLevel :=
           BeforeConsequencesLevel
    AfterRTRMRiskLevel :=
           BeforeRTRMRiskLevel
    COMMENT main RTRM risk management loop
     WHILE AfterRTRMRiskLevel > RiskAcceptanceLevel
      AfterRTRMCountermeas :=
           SelectBetterCountermeasures()
      AfterRTRMLikelihoodLevel := AssessLikelihood()
      AfterRTRMConsequencesLevel :=
           AssessConsequences()
      AfterRTRMRiskLevel :=
           AfterRTRMLikelihoodLevel
           *AfterRTRMConsequencesLevel
     ENDWHILE
   ENDFOR (vulnerabilities)
  ENDFOR (threats)
 ENDFOR (assets)
ENDIF (RTRM_event)
```

Table 1. Real-time risk management process – inputs and activities

Source	Description	Invoked	Identification time	Nature	Main output	RTRM activities
SEM	Real-time monitoring of the IT system behaviour (events), the correlations analysis of events, issuing warnings and aggregated information	By behaviour anomalies	Fast	Ex ante	Threat symptoms Possible vulnerability	Find relevant assets and perform the RTRM process with respect to the <threat, vulnerab.>
SIEM	Long term storage, analysis, reporting of log data, issuing aggregated alerting data	By behaviour anomalies	Fast	Ex ante	Threat symptoms Possible vulnerability	Find relevant assets and perform the RTRM process with respect to the pair <threat, vulnerab.>
Incident management	Identifies causes, consequences and circumstances of the occurred incident	On incident	Fast	Ex post	Breached asset, Damages, Occurred threat, Exploited vulnerability	Perform RTRM process for relevant assets using available information
Penetration tests and vulnerability scanners	Controlled attack on a computer system that looks for security weaknesses, potentially gaining access to the computer features and data	On request	Fast	Ex ante	Not properly countered threats, Possible vulnerabilities	Find relevant assets and perform the RTRM process with respect to the <threat, vulnerab.>
Audit process	Identifies non compliances including exposures to new threats, new vulnerabilities, non-efficient countermeasures, organizational and procedural shortcomings, etc.	On request	Slow	Ex ante	Not properly countered threats Possible vulnerabilities Possible damages	Perform RTRM process for relevant assets using available information

(*continued*)

Table 1. (*continued*)

Source	Description	Invoked	Identification time	Nature	Main output	RTRM activities
Security effectiveness measures	Provide aggregating security-related characteristics	On warning (issued when a measure exceeds the assumed level)	Slow	Ex ante	Breached asset, Damages, Occurred threat, Exploited vulnerability, Incident rates, Protection costs, Security management status	Perform RTRM process for relevant assets using available information

Summing up, for any triple <asset, threat, vulnerability> belonging to the assessed area, the risk before and after reduction is assessed and the existing and applied countermeasures are specified as the real-time risk record:

```
rtr=(asset, threat, vulnerability, BeforeRTRMCountermeas, BeforeRTRMLikelihoodLevel, BeforeRTRMConsequencesLevel, BeforeRTRMRiskLevel, AfterRTRMCountermeas, AfterRTRMLikelihoodLevel, AfterRTRMConsequencesLevel, AfterRTRMRiskLevel).
```

The real-time risk can be considered as a sum of these records for all possible RT scenarios:

$$RTR = \bigcup_{all\,RT\,scenarios} \{rtr\} \tag{2}$$

The RTRM risk assessment creates a dynamic picture of the risk situation during a certain SRA period i. This picture consists of a set of scenarios in the area impacted by changes of the risk relevant factors. These moments can be shown as points on a discrete time scale too, in the range of the given SRA period i: $RTR(T_{i,0})$, $RTR(T_{i,1})$, $\ldots RTR(T_{i,j})$, $RTR(T_{i,j+1})$, $RTR(T_{i,j+2})$, $\ldots RTR(T_{i,m})$, where $RTR(T_{i,m})$ is the real-time risk accumulated at the end of SRA period i (*m* RTRM events occurred). Please note that $RTR(T_{i,j})$ is determined using the risk factors identified when `RTRM_event` j occurs.

3.3 Mixed-Mode Risk Management Process

In the pure static approach, $SRA(T_{i+1})$ is determined with the use of risk factors identified at the end of $SR(T_i)$ and it represents the current static risk picture, valid until the new analysis result, i.e. $SR(T_{i+2})$ replaces the old one. SRA is focused on steady state or low frequency risk factors. The real-time-risk management process is performed parallel to the static risk management during the given SRA period and is focused on dynamically changing, fluctuating risk factors. The RTRM process can be invoked many times during this period and particular events may concern different areas of the risk picture. The RTRM process samples dynamically occurring events and reacts to them. The dynamic risk picture cumulates all dynamic risks within the SRA period.

The authors propose a new integrated approach combining the static, real-time and adaptive approaches.

It is assumed that at the beginning of the SRA period i, a real-time risk analyzer is initialized with the current static assessment results: $RTR\left(T_{i,0}\right) = SR(T_i)$. This ensures a common starting point of both assessments. Starting from this point the real-time risk is cumulated until the end of period i. The accumulated result $RTR\left(T_{i,m}\right)$ is used in the adaptive risk assessment for the next period i + 1 (SRA and ARA periods are the same).

When the SRA/ARA period ends, the dynamic risk picture parameters are considered by the new ARM process allowing to determine the adaptive risk $AR(T_{i+1})$:

$$AR(T_{i+1}) = ARF\left[SR(T_{i+1}),\ RTR\left(T_{i,m}\right)\right] \tag{3}$$

The new SRA results, i.e. $SR(T_{i+1})$, are refined by the RTR results $RTR\ \left(T_{i,m}\right)$ accumulated at the end of the previous period. This refinement, expressed by the *Adaptive Risk Function* (*ARF)*, is a complex heuristic process in which many fuzzy factors and relations must be taken into consideration. Three groups of data are used on the input to determine the adaptive risk for the next period $(AR(T_{i+1}))$:

- the static risk picture at the beginning of the previous SRA/ARA period $(SR(T_i) = RTR\left(T_{i,0}\right))$: `BeforeCountermeas, BeforeLikelihoodLevel, BeforeConsequencesLevel, BeforeRiskLevel`; it expresses the common reference point, the context of assessment;
- the updated static risk picture at the beginning of the next SRA/ARA period $(SR(T_{i+1}))$: `AfterCountermeas, AfterLikelihoodLevel, AfterConsequencesLevel, AfterRiskLevel`; it represents the risk related to steady state or low frequency risk factors;
- the cumulated real-time risk $\left(RTR\left(T_{i,m}\right)\right)$ at the end of the previous SRA/ARA period expresses the "corrections" caused by real symptoms or incidents occurred and the knowledge related to them: `AfterRTRMCountermeas, AfterRTRMLikelihoodLevel, AfterRTRMConsequencesLevel, AfterRTRMRiskLevel`; it represents dynamically changing, fluctuating risk factors.

It is necessary to answer some difficult questions, like: how to merge the static and dynamic countermeasures, how they will impact jointly threats and vulnerabilities, how to assess the risk level and consequences in this situation, etc. The automation of this decision process may be advantageous though it still remains a challenge.

The above described mixed-mode risk management process focused on the selected risk scenario is summarized in Fig. 1. Please note the static and real-time paths and the process adaptation (a decision improvement by RTR acquired data) before the next step. Please note that only the most important elements of risk records are shown.

4 Exemplification of Mixed-Mode Risk Management Process

The exemplification of the presented concept is restricted to the selected example embracing one asset A_x. A more extensive validation is planned after the methodology implementation on the software platform.

It is assumed that likelihood levels and consequences levels (SR/RTR/AR) are measured in the range from 0 to 10, causing the risk range from 0 to 100. Risk acceptance levels are 60.

Let us consider a scenario: Threat T_y exploiting the vulnerability V_z breaches the asset A_x despite of the existing countermeasure C_p. It is also assumed that every countermeasure represents a set of different, working coherently security measures: technical, physical and organizational. A short notation for risk records, instead of self-explaining ones, is proposed as well:

SCM_m=BeforeCountermeas, SCM_{m+1}=AfterCountermeas,
$RTCM_n$=BeforeRTRCountermeas, $RTCM_{n+1}$=AfterRTRCountermeas.

1. The first static risk assessment (T_1), started with $SCM_{1,0}=C_p$ is performed on request and gives the following results: $SR(T_1)=(SCM_0, 70, SCM_1, 55)$, where BeforeRiskLevel=70, AfterRiskLevel=55. These data are used to initialize the real-time risk record $RTR(T_{1,0})=SR(T_1)$.
2. Let us assume that during the T_1 period an incident related to the considered threat T_y occurs (RTR_event). The incident lessons learnt conclude that the likelihood and consequences (i.e. the risk) were underestimated (probably they are higher than 55) in reality. In order to maintain the risk on this level, better countermeasures should be applied (to reduce vulnerability/threat impact). The existing countermeasures SCM_1 are improved to $RTCM_2$, causing the risk level modification: $RTR(T_{1,1}) = (RTCM_0 = SCM_1, 55, RTCM_1, 54)$. The RTR risk level is a little lower (54), because $RTCM_1$ is more effective than $RTCM_0$.
3. Let us assume that during T_1 SIEM discovers an anomaly in the area related to T_y (RTR_event), interpreted as attack symptoms. The security problem analysis concludes that certain risk scenarios related to these symptoms, including the considered scenario, should be reassessed. As a result of this RTR reassessment, the countermeasures were improved again: $RTR(T_{1,2}) =(RTCM_1, 54, RTCM_2, 53)$ and RTR risk level decreased (54–>53).
4. Let us assume that during T_1 the audit in the data center detects a previously unknown vulnerability (RTR_event), reinitiating the RTRM process. The

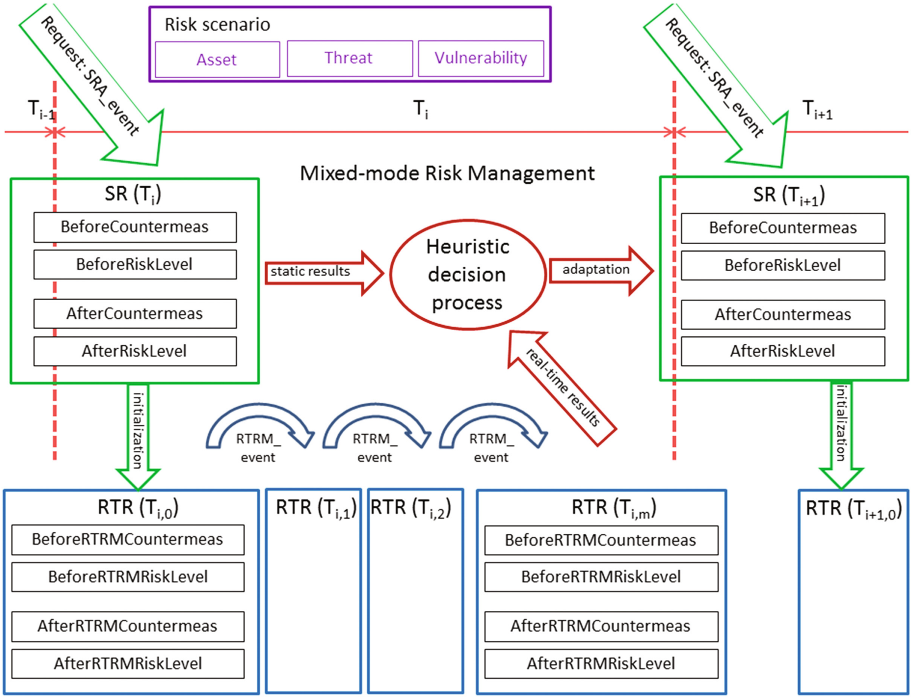


Fig. 1. General scheme of the mixed-mode risk management

reassessment concludes that this vulnerability should be removed by further countermeasures improvement ($\mathtt{RTCM}_3$): $\mathtt{RTR}(\mathtt{T}_{1,3}) = (\mathtt{RTCM}_2, 53, \mathtt{RTCM}_3, 52)$.

5. The security policy statement regulates the times of the next static risk assessment which has to be done. The $\mathtt{SR}(\mathtt{T}_2)$ assessment takes into consideration the current risk situation, including all RTR assessments and countermeasures applied during T_1 (an adaptive approach): $\mathtt{SR}(\mathtt{T}_2) = \mathtt{AR}(\mathtt{T}_2) = \mathtt{ARF}[\mathtt{SR}(\mathtt{T}_1), \mathtt{RTR}(\mathtt{T}_{1,3})]$.

The $\mathtt{SR}(\mathtt{T}_2)$ assessment results are used to initialize $\mathtt{RTR}(\mathtt{T}_{2,0})$. The adaptive risk management system starts to watch for RTR_events during the next time period T_2.

5 Conclusions

The paper is focused on the integration of static and dynamic risk management methods to minimize drawbacks, when they are used separately. The mixed-mode risk management method is proposed.

The static approach is comprehensive and embraces relatively long time periods, e.g. one year, and its loop is invoked on the security managers' request. Between static risk assessments the fluctuating risk factors (new threats, their symptoms, unconformities, new vulnerabilities, etc.) may occur suddenly and countermeasures are not able to counter them. Thus during these periods the real-time risk management loop is invoked on each dynamically occurred event, allowing for the reaction and correction of countermeasures. The security related data sampled by the RTR manager during this period are used to update the risk picture for the next "static" time period. Section 4 presents an example how the mixed-mode approach can be used.

The subject of discussion is very extensive because it comprises the integration of several subsystems, like risk manager, incident manager, SIEM, audit subsystem, etc. Most of them exist in the OSCAD software platform [15]. For this reason a more comprehensive validation is planned using this platform. This will allow to reach a certain level of automation. Future research should embrace also the unified representation of threats, vulnerabilities, assets, countermeasures and relations between them within the mixed-mode risk manager using renown standards.

References

1. ISO 31000:2009, Risk management – Principles and guidelines
2. ISO/IEC 31010:2009 – Risk Management – Risk Assessment Techniques
3. Rausand, M.: Risk Assessment: Theory, Methods, and Applications. Statistics in Practice (Book 86). Wiley (2011)
4. Hokstad, P., Utne, I.B., Vatn, J. (eds.): Risk and Interdependencies in Critical Infrastructures: A Guideline for Analysis. Reliability Engineering. Springer-Verlag, London (2012). doi:10.1007/978-1-4471-4661-2_2
5. Bialas, A.: Risk management in critical infrastructure—foundation for its sustainable work. Sustainability **8**, 240 (2016). http://www.mdpi.com/2071-1050/8/3/240/htm
6. ENISA. http://rm-inv.enisa.europa.eu/methods. Access date: Dec 2016
7. Bjerga, T., Aven, T.: Adaptive risk management using new risk perspectives – an example from the oil and gas industry. Reliab. Eng. Syst. Saf. **134**, 75–82 (2015)
8. Oltsik, J.: Real-Time Risk Management (2010). http://la.trendmicro.com/media/misc/real-time-risk-management-en.pdf. Access date: Nov 2016
9. NetIQ Sentinel. https://www.netiq.com/products/sentinel/. Access date: Jan 2017
10. QRadar IBM. http://searchsecurity.techtarget.com/feature/IBM-Security-QRadar-SIEM-product-overview. Access date: Jan 2017
11. Marvell, S.: The real and present threat of a cyber breach demands real-time risk management, Acuity Risk Management (2015)
12. TAXII. http://taxiiproject.github.io/about/. Access date: Jan 2017
13. MITRE. https://cve.mitre.org/cve/. Access date: Jan 2017
14. Dulac, N.: A Framework for Dynamic Safety and Risk Management Modeling in Complex Engineering Systems, Ph. D. thesis, Massachusetts Institute of Technology (2007). http://sunnyday.mit.edu/safer-world/dulac-dissertation.pdf. Access date: Mar 2017
15. OSCAD project. http://www.oscad.eu/index.php/en/. Access date: Dec 2016

Software Support of the Common Criteria Vulnerability Assessment

Andrzej Bialas(✉)

Institute of Innovative Technologies EMAG,
Leopolda 31, 40-189 Katowice, Poland
andrzej.bialas@ibemag.pl

Abstract. The paper deals with the Common Criteria assurance methodology, particularly vulnerability assessment which is the key activity of the IT security evaluation process. Vulnerability assessment is specified by the Common Criteria Evaluation Methodology (CEM). The paper is focused on software support for vulnerability assessment. As the implementation platform, a ready-made risk management software developed by the author's organization is applied. The paper includes introduction to the vulnerability assessment, review of the existing methods and tools, specification of the CEM-based method to be implemented in the software, implementation and short exemplification. The conclusions summarize the validation and propose future works to extend and improve the tool.

Keywords: Common criteria · Vulnerability assessment · Attack potential · Risk management · IT security development · Security assurance

1 Introduction

Today's societies and economies are based on Information and Communication Technologies (ICT). Digitalization increases cyber security risk across many sectors. There is a necessity to minimize the risk inherent to the ICT use in the society and economy. One of the ways to do it is rigorous development of ICT products as accompanied by their independent evaluation and certification. This approach allows to achieve security assurance for the products, still, it is complex, time and cost consuming. Computer support for developers' or evaluators' works, focused on the most difficult or arduous activities, allows to make the development and evaluation processes much more efficient.

The paper concerns computer support for the Common Criteria compliant IT security development and assessment process. The Common Criteria (CC) [1] methodology presented in the ISO/IEC 15408 standard is a basic assurance methodology. According to this methodology, the assurance is measurable using EALs (Evaluation Assurance Levels) in the range EAL1 to EAL7. The CC methodology comprises three basic processes:

- IT security development process of the IT product or system called TOE (Target of Evaluation); after different security analyses have been performed, a document is

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_3

prepared, called Security Target (ST); the Security Target embraces the security problem definition (SPD), its solution by specifying security objectives (SO), security requirements and functions; security functional requirements (SFRs), derived from the security objectives, describe how security measures should work in the operational environment; security assurance requirements (SARs), related to the EAL, determine how much assurance we can have in an IT product; the ST includes TOE security functions (TSF) meeting SFRs; the TSFs are implemented on the claimed EAL;

- TOE development process (according to the EAL) concerns the IT products (TOE) and their documentation; they are submitted together with the ST to the security evaluation process;
- security evaluation process carried out in an independent, accredited laboratory in a country which is a signatory of the Common Criteria Recognition Arrangement (CCRA) [2].

The Common Criteria methodology is described in publications worldwide [3, 4] and the author's publications [5–8].

A very important issue of the Common Criteria methodology is vulnerability assessment whose rigour depends on the claimed EAL. Vulnerabilities can be exploited by threat agents in the product operational environment. This exploitation results in security breaches. The development process should be focused on the vulnerabilities minimization. The developer, specifying the threat environment (SPD), should consider the attack potential, i.e. a measure of the effort to be expended in attacking a TOE, expressed in terms of the attacker's expertise, resources and motivation [1]/part1.

In addition, vulnerability assessment is one of the key elements of evaluation. The vulnerability assessment is performed according to AVA_VAN (Vulnerability analysis assurance components family). The AVA_VAN assurance requirements (components) are specified in the third part of the Common Criteria standard [1] and the CEM (Common Evaluation Methodology) document [9]. Annex B of this document [9] includes a general guideline for evaluators. They are able to see how to calculate the attack potential possessed by the assumed attackers (threat agents) of the TOE. The methodology presented in the paper complies with these guidelines. Please note that the vulnerability assessment is a specific risk assessment focused on attack potential, expressing possibility of the asset breach. This possibility is related to the term "likelihood" used in the risk assessment. The second factor – "consequences" is considered but not explicitly assessed.

The objective of the paper is to present a software implementation of the Common Criteria vulnerability assessment methodology. The Enhanced Risk Manager (ERM) developed by the author's organization is used as the software implementation platform.

Section 2 discusses the vulnerability assessment issue. Section 3 features the review of publications in the paper domain. The review shows that there are no works related to the automation of the vulnerability assessment. Section 4 presents the methodology, and Sect. 5 its implementation. The paper is completed by conclusions – Sect. 6.

2 Vulnerability Assessment According to Common Criteria

Vulnerability assessment (VA) is focused on the flaws or weaknesses in the TOE working in its operational environment. First, they should be identified and then assessed whether they can be exploited by threats agents of the defined capability, i.e. the attack potential sufficient to violate the SFRs.

The assessment embraces vulnerabilities possible to exploit by a huge number of attacks, which can be ordered by five main categories: bypassing, tampering, direct attack, monitoring and misuse [9]. Each category has its subcategories.

Bypassing embraces means which can be used by an attacker to avoid security enforcement done by TSFs. Bypassing may concern:

- inheriting privileges or other capabilities that would not be granted otherwise;
- exploiting the capabilities of TOE interfaces, or of utilities interacting with the TOE;
- getting access to sensitive data stored in or copied to inadequately protected areas, where confidentiality is a concern.

Tampering encompasses attacks focused on the behaviour of the TSF (i.e. corruption or deactivation), for example:

- forcing the TOE to deal with unusual or unexpected circumstances;
- disabling or delaying security enforcement;
- getting access to data whose confidentiality or integrity the TSF relies on;
- modifying the TOE in its physical aspect.

Direct attack covers the identification of penetration tests to check the strength of permutational or probabilistic mechanisms and other mechanisms to ensure they withstand a direct attack.

Monitoring attack is aimed at information related to the TOE operations, e.g.: information of internal TOE transfer or export from the TOE, information generated and passed to other user data or gained through monitoring the operations.

Misuse may be caused by:

- incomplete guidance documentation;
- unreasonable guidance;
- forced exception behaviour of the TOE;
- unintended misconfiguration of the TOE.

This classification has an open character and is refined to express specific technological issues related to the given IT product category.

The examples of vulnerabilities are:

- absence of the required security enforcement on interfaces or utilities,
- possibility to illicitly acquire privileges or capabilities of a privileged component,
- inadequately protected areas.

3 State of the Art in the Research Domain

The review embraces the following fields:

- methods and tools supporting the CC methodology processes, including evaluation,
- vulnerability assessment methods and tools.

Apart from the Common Criteria standard and the supplementing guidelines [2], the support given to the Common Criteria methodology developers and evaluators is rather poor and embraces only the following:

- general guidelines, like: ISO/IEC 15446 [10] for security targets and protection profiles elaboration, the BSI guide [11] for other evidences up to EAL5, and books, like [3, 4], however, they are focused on the Common Criteria methodology presentation, not on vulnerability assessment;
- a few software tools, like: Common Criteria (CC) ToolboxTM [12], GEST [13], Trusted Labs Security Editing Tool (TL SET) [14]; please note that these tools are focused on the ST preparation and do not support the elaboration of other evidences concerning the TOE design, life cycle, guidance, testing, vulnerability assessment, etc.; they do not deal with vulnerability assessment either;
- the CCMODE Tools [15] embraces full implementation of CEM, but omits vulnerability assessment details discussed in this paper;
- the extensive Information Assurance Technology Analysis Center (IATAC) [16] report encompasses detailed methods and tools focused on different kinds of vulnerability analyses designed for specific IT products or systems, like network scanners, host scanners, web application scanners, multilevel scanners, penetration test tools, vulnerability scan consolidators, etc.; they can be used on the lower layer of the software tool presented in this paper.

There are no specialized software tools supporting vulnerability assessment.

4 Methodology

During the vulnerability assessment a huge number of attack scenarios are assessed with respect to potential vulnerabilities. Some vulnerabilities are encountered during evaluation activities "by the way", some of them are results of unstructured, focused or even methodical analyses. The rigour applied depends on the claimed EAL.

For all exploitable vulnerabilities the evaluator calculates the attack potential to determine whether the exploitation conditions are adequate to the level of the attack potential assumed for the attacker. The attack potential rises proportionally to the increasing motivation, resources and expertise of the attacker.

The attack potential (AP) sufficient to breach the TOE can be expressed as the sum of factors:

$$AP = ET + SE + KT + WO + EQ, \text{ where:} \tag{1}$$

ET – Elapsed Time, i.e. time taken to identify and exploit the vulnerability (Table 1);

SE – Specialist Expertise, i.e. the level of generic knowledge of the attacker (Table 2);

Table 1. Measures of the Elapsed Time (ET)

Elapsed time	ET	Elapsed time	ET
<=1 day	0	<=3 months	10
<=1 week	1	<=4 months	13
<=2 weeks	2	<=5 months	15
<=1 month	4	<=6 months	17
<=2 months	7	>6 months	19

Table 2. Measures of the Specialist Expertise (SE)

Specialist expertise	SE	Comments
Laymen	0	Has no particular expertise
Proficient	3	Is familiar with the security behaviour of the product or system type
Expert	6	Is familiar with the underlying algorithms, protocols, hardware, structures, security behaviour, principles and employed concepts of security, techniques and tools for defining new attacks, cryptography, classical attacks for the product type, attack methods, etc. – implemented in the product or system type
Multiple expert	8	On the expert level different fields of expertise are required for distinct steps of the attack

Table 3. Measures of the Knowledge of the TOE (KT)

Knowledge of the TOE	KT	Example
Public	0	Information gained from the Internet
Restricted	3	Knowledge controlled within the developer's organisation and shared with other organizations under a non-disclosure agreement
Sensitive	7	Knowledge that is shared between discreet teams within the developer's organisation. Only members of specified teams have access to it
Critical	11	Knowledge familiar only to a few individuals. The access to this knowledge is very strictly controlled on a strict-need-to-know basis and individual undertaking

KT – Knowledge of the TOE, i.e. dealing with the TOE design, operation (Table 3);
WO – Window of opportunity, i.e. considerable amounts of access to the TOE or the number of samples of the TOE that the attacker can obtain; related to ET (Table 4);
EQ – Equipment, i.e. IT hardware/software or other equipment required to identify or exploit the vulnerability (Table 5).

Table 4. Measures of the window of opportunity (WO)

Window of opportunity	WO	Example
Unnecessary/unlimited access	0	The attack does not need any kind of opportunity to be performed because there is no risk it will be detected during access to the TOE and one can access the number of TOE samples for the attack
Easy	1	Access is required for less than a day. Fewer than 10 TOE samples are required to perform the attack
Moderate	4	Access is required for less than a month. Fewer than one hundred TOE samples are required to perform the attack
Difficult	10	Access is required for at least a month. At least one hundred TOE samples are required to perform the attack
None	**	The attack path is not exploitable due to other measures in the intended operational environment of the TOE; the period during which the asset to be exploited is available or is sensitive is shorter than the opportunity period needed to perform the attack

Table 5. Measures of the equipment (EQ)

Equipment	EQ	Example
Standard	0	Is readily available to the attacker, either to identify a vulnerability or to attack. This equipment may be a part of the TOE as such (e.g. a debugger in an operating system), or can be readily obtained (e.g. Internet downloads, protocol analyser or simple attack scripts)
Specialised	4	Is not readily available to the attacker, but could be acquired without too much effort. This could include the purchase of some equipment (e.g. power analysis tools, use of hundreds of PCs linked across the Internet), or development of more extensive attack scripts or programs. If distinct steps of an attack require clearly different test benches consisting of specialised equipment, this would be rated as bespoke
Bespoke	7	Is not readily available to the public as it may need to be specially produced (e.g. very sophisticated software), or because the equipment is so specialised that its distribution is controlled, or even restricted. The equipment may be also very expensive
Multiple bespoke	9	Different types of bespoke equipment are required for distinct steps of an attack

Table 6. Attack potential and the TOE resistance

AP range	Attack potential required to exploit scenario	TOE resistant to attackers with attack potential of
0–9	Basic	No rating
10–13	Enhanced-Basic	Basic
14–19	Moderate	Enhanced-Basic
20–24	High	Moderate
=>25	Beyond high	High

All factors are measured using the predefined enumerative scales (Tables 1, 2, 3, 4 and 5).

The symbol specified in Table 4 "**" has special meaning and should not be seen as natural progression from the timescales specified in the preceding ranges associated with this factor.

The AP is assessed in the range 0 to 57 with special WO = None option. Table 6, based on CEM [9] (Annex B/Table 4), shows how these values are assigned to the enumeration values (second and third columns). For example, it means that if the minimal AP to exploit scenarios is 15, i.e. "Moderate", than the TOE is resistant to attackers with the attack potential of "Enhanced-Basic".

The above method, compatible with those specified in this annex, is sufficient to calculate the attack potential. This method does not go beyond the vulnerability assessment, e.g. towards risk assessment/management, though it is related to this issue.

5 Implementation and Validation

The ERM configurable software platform has been incrementally developed. Currently it contains the following risk assessment methods [17, 18] implemented:

- Consequences-probability matrix (called here TVC, because it is based on triples: threat-vulnerability-controls),
- Business impact analysis (BIA),
- Fault Tree Analysis (FTA),
- Event Tree Analysis (ETA).

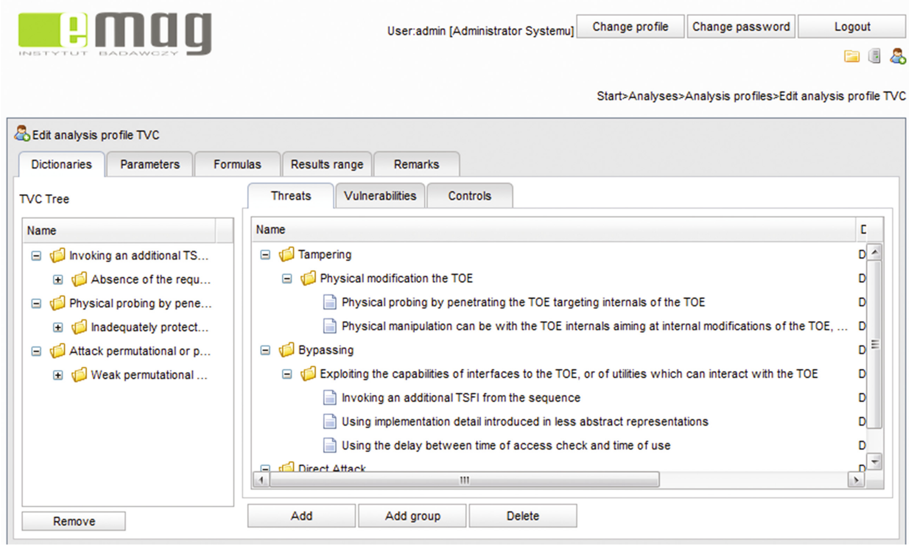
For each kind of analysis one or more analysis profiles can be defined, embracing dictionaries, formulas and configuration parameters. Using the given profile, one or more analyses can be provided, and based on the risk assessment results, new or improved security measures can be proposed (in this sense it is a risk management tool). This tool has an open character and it should be validated in different application domains. The paper [19] shows the tool and one of the first ERM applications.

Apart from the first aim, i.e. providing the ERM-based vulnerability assessment tool (ERM-VA) for Common Criteria evaluators, the second aim is to validate the ERM software itself to sample data for its future development. ERM-VA is based on the TVC method. The implementation embraces the profile elaboration and performing analyses. The goals of the implementation are the following:

- to manage the systematic and complex vulnerability assessment process,
- to calculate attack potentials for considered attack scenarios and to assess the TOE resistance to the attacks of the given potential.

5.1 TVC-VA Analysis Profile

The TVC-VA profile should consider the predefined threats and vulnerabilities discussed in CEM [9] (Annex B/Table 4). Analyzing this document, the lists of predefined threats and vulnerabilities are elaborated and introduced to the TVC-VA profile. The main categories and subcategories of threats and the list of vulnerabilities are presented in Sect. 2. No controls, i.e. security measures are needed for the vulnerability analysis. The dictionaries are open. They can be refined or supplemented for the assessed IT product to express its specific character. The right part of Fig. 1 shows a part of the global threat dictionary made according to CEM/Annex B including two main categories: bypassing and tampering with subcategories and particular threats. The left part shows three selected pairs threat-vulnerability for further analysis Fig. 2.

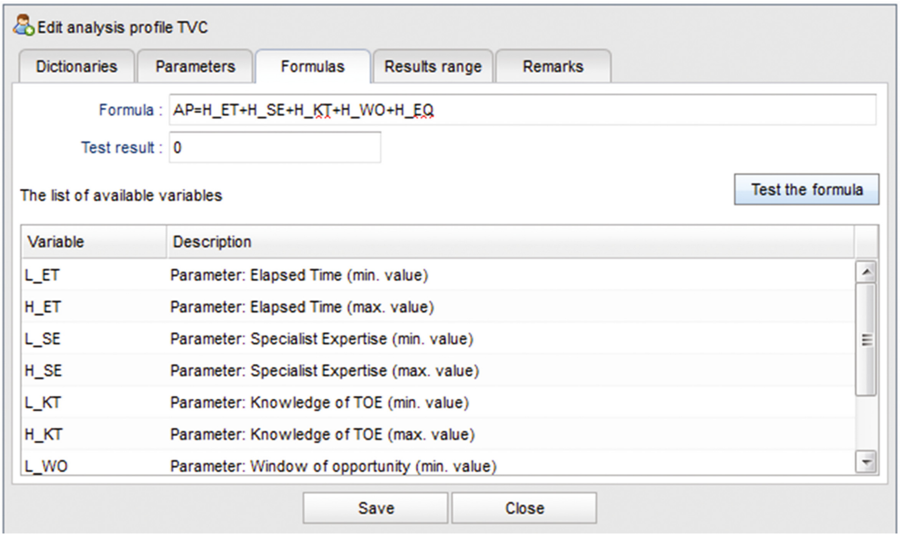


- Invoking an additional TS...
  - Absence of the requ...
- Physical probing by pene...
  - Inadequately protect...
- Attack permutational or p...
  - Weak permutational ...

- Tampering
  - Physical modification the TOE
    - Physical probing by penetrating the TOE targeting internals of the TOE
    - Physical manipulation can be with the TOE internals aiming at internal modifications of the TOE, ...
- Bypassing
  - Exploiting the capabilities of interfaces to the TOE, or of utilities which can interact with the TOE
    - Invoking an additional TSFI from the sequence
    - Using implementation detail introduced in less abstract representations
    - Using the delay between time of access check and time of use
- Direct Attack

Fig. 1. Threat dictionary – an example in the EMAG-ERM tool

Each pair represents an attack scenario (AS). For the given AS the attack potential should be calculated. Thus the basic formula (Eq. 1) used in Common Criteria is implemented in the tool. The ERM formulas and their variables are defined by the user. Each variable has minimal (L_VAR) and maximal (H_VAR) values to express uncertainties. For the discussed profile, uncertainties are not considered, both values are



Formula : AP=H_ET+H_SE+H_KT+H_WO+H_EQ

Test result : 0

The list of available variables

Variable	Description
L_ET	Parameter: Elapsed Time (min. value)
H_ET	Parameter: Elapsed Time (max. value)
L_SE	Parameter: Specialist Expertise (min. value)
H_SE	Parameter: Specialist Expertise (max. value)
L_KT	Parameter: Knowledge of TOE (min. value)
H_KT	Parameter: Knowledge of TOE (max. value)
L_WO	Parameter: Window of opportunity (min. value)

Fig. 2. Attack potential formula implemented in the TVC-VA profile in the EMAG-ERM tool

the same and only variables with prefixes "H_" are used. Each variable representing an AP factor has predefined measures according to Table 1 through Table 5.

Figure 3 shows an example of the factor measure configuration. It is an implementation of Table 5 concerning the threat agent equipment.

This way the TVC-VA profile is ready to use.

5.2 TVC-VA Analysis

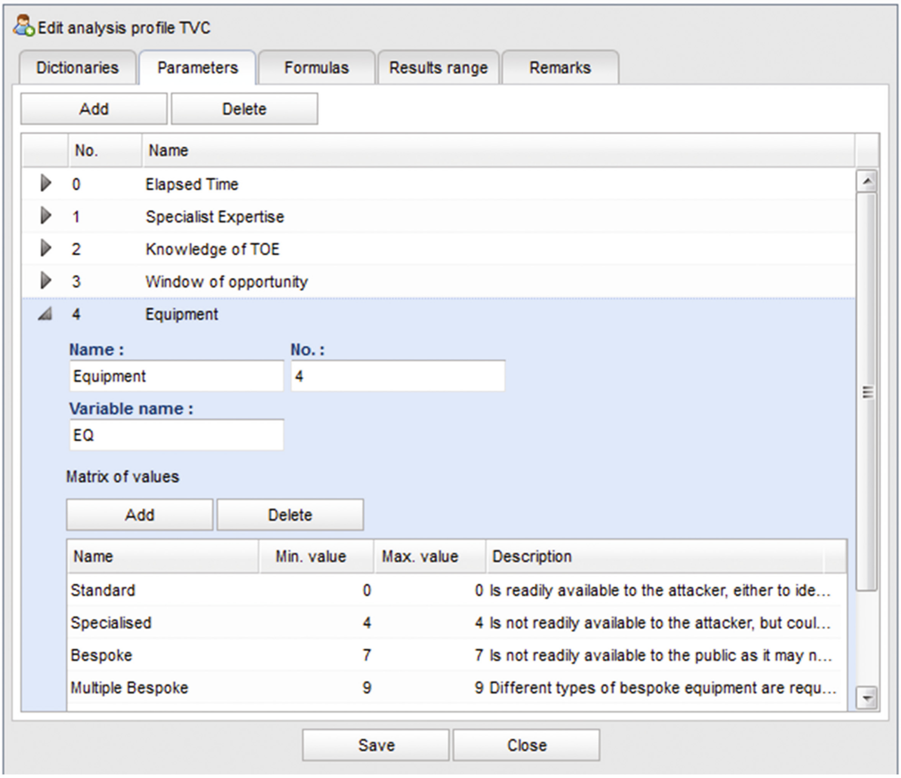
The vulnerability assessment is shown by an example embracing three attack scenarios (Fig. 4). The left part of the window presents these scenarios, the right part the assessment of particular factors with respect to the first AS:

- threat: "Invoking an additional TSFI from the sequence",
- vulnerability: "Absence of the required security enforcement on interfaces or utilities".

Each factor has basic and extended explanations helping the evaluator.

Figure 5 presents the results of the assessment. Different attack potentials are obtained, belonging to different enumerative values (please compare these results with Table 6).

Based on this table the following conclusion is possible. One of the attack scenarios has AP = 18. It means that the considered TOE can be exploited by an intruder of this attack potential, so the TOE is resistant to the attackers with attack potential of "Enhanced-Basic". Particular attack potentials are marked by different colours.



No.	Name
0	Elapsed Time
1	Specialist Expertise
2	Knowledge of TOE
3	Window of opportunity
4	Equipment

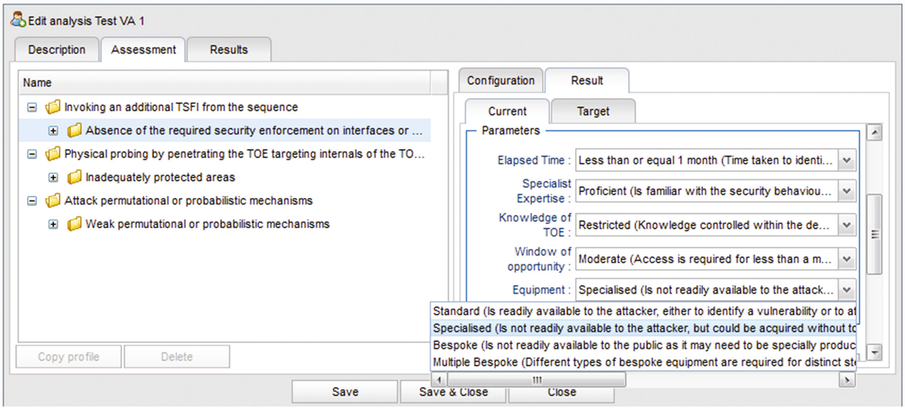
Name : Equipment

No. : 4

Variable name : EQ

Name	Min. value	Max. value	Description
Standard	0	0	Is readily available to the attacker, either to ide...
Specialised	4	4	Is not readily available to the attacker, but coul...
Bespoke	7	7	Is not readily available to the public as it may n...
Multiple Bespoke	9	9	Different types of bespoke equipment are requ...

Fig. 3. Equipment (EQ) factor related to data in the TVC-VA profile in the EMAG-ERM tool



Name
- Invoking an additional TSFI from the sequence
  - Absence of the required security enforcement on interfaces or ...
- Physical probing by penetrating the TOE targeting internals of the TO...
  - Inadequately protected areas
- Attack permutational or probabilistic mechanisms
  - Weak permutational or probabilistic mechanisms

Elapsed Time : Less than or equal 1 month (Time taken to identi...

Specialist Expertise : Proficient (Is familiar with the security behaviou...

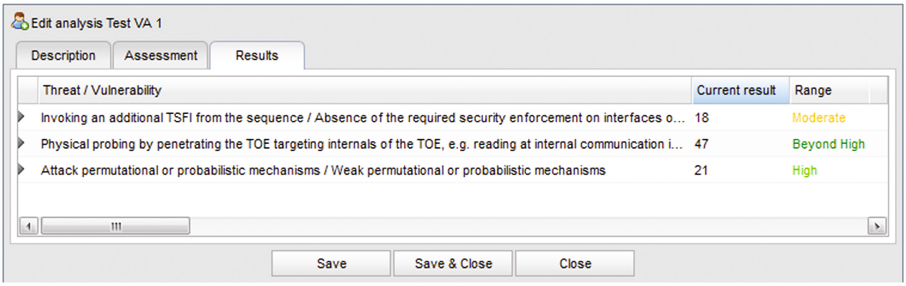
Knowledge of TOE : Restricted (Knowledge controlled within the de...

Window of opportunity : Moderate (Access is required for less than a m...

Equipment : Specialised (Is not readily available to the attack...

Standard (Is readily available to the attacker, either to identify a vulnerability or to at
Specialised (Is not readily available to the attacker, but could be acquired without to
Bespoke (Is not readily available to the public as it may need to be specially produc
Multiple Bespoke (Different types of bespoke equipment are required for distinct st

Fig. 4. Vulnerability assessment based on the TVC-VA profile in the EMAG-ERM tool



Threat / Vulnerability	Current result	Range
Invoking an additional TSFI from the sequence / Absence of the required security enforcement on interfaces o...	18	Moderate
Physical probing by penetrating the TOE targeting internals of the TOE, e.g. reading at internal communication i...	47	Beyond High
Attack permutational or probabilistic mechanisms / Weak permutational or probabilistic mechanisms	21	High

Fig. 5. Vulnerability assessment results in the EMAG-ERM tool

6 Conclusions

The paper is focused on two main goals:

- automation of arduous and difficult vulnerability assessment,
- sampling experience for further development of the Enhanced Risk Manager software.

Analyzing CEM, especially in the range of vulnerability assessment, a concise specification of the vulnerability assessment method is prepared, being an input for software implementation of the method on the open, fully configurable ERM risk manager. The implementation embraces the analysis profile definition according to the vulnerability assessment methodology, i.e. preparing dictionaries of threats and vulnerabilities, defining some attack scenarios for analysis, defining the formula and measures of the formula factors. For the given profile many analyses can be performed, including the one presented in the paper.

For each identified attack scenario, i.e. the pair threat-vulnerability, the attack potential is calculated and classified according to CEM as: “Basic”, “Enhanced-Basic”, “Moderate”, “High”, “Beyond High”. It allows to determine the TOE resistance to attackers with the given attack potential.

The quantitative values of the attack potential are automatically translated to commonly used enumeration values and marked by colours. The validation shows that the complete vulnerability assessment can be performed using the ERM tool.

One issue was not properly implemented in the ready-made software. It concerns the “exception” in the Window of opportunity factor, i.e. WO = NONE. The possibility of the implementation exists, but needs extra, uncomplicated programming (introducing a special flag to mark this exception).

The threat and vulnerability dictionaries fully implement items extracted from CEM [9] (Annex B), but it should be noted that they have a generic character. They should be extended by IT products/technologies specific items or specified within the supporting guides and standards, like [20].

The discussed software support embraces the main management activities of vulnerability assessment. Please note that each attack scenario represents an experiment, sometimes very complex, long lasting and based on specialized equipment. The tool is able to manage results of these experiments, but particular experiments should be performed using the specific methods and tools (similar to these, owned by potential attackers). This range of automation seems to be enough. A good idea would be to integrate the Common Criteria evaluation tool existing in the CCMODE Tools [15] with the presented vulnerability assessment tool discussed in the paper.

References

1. Common Criteria for IT Security Evaluation, part 1–3, version 3.1 rev. 4 (2012). http://www.commoncriteriaportal.org/. Accessed 10 Mar 2017
2. Common Criteria Portal Home page. http://www.commoncriteriaportal.org/. Accessed 10 Mar 2017
3. Hermann, D.S.: Using the Common Criteria for IT Security Evaluation. CRC Press, Boca Raton (2003)
4. Higaki, W.H.: Successful Common Criteria Evaluation. A Practical Guide for Vendors. Copyright 2010 by Wesley Hisao Higaki, Lexington (2011)
5. Bialas, A.: Intelligent sensors security. Sensors **10**, 822–859 (2010)
6. Bialas, A.: Common criteria related security design patterns—validation on the intelligent sensor example designed for mine environment. Sensors **10**, 4456–4496 (2010)
7. Bialas, A.: Common criteria related security design patterns for intelligent sensors—knowledge engineering-based implementation. Sensors **11**, 8085–8114 (2011)
8. Bialas, A.: Computer-aided sensor development focused on security issues. Sensors **16**, 759. http://www.mdpi.com/1424-8220/16/6/759. Accessed 10 Mar 2017
9. Common Methodology for Information Technology Security Evaluation, version 3.1 rev. 4 (2012). http://www.commoncriteriaportal.org/. Accessed 10 Mar 2017
10. ISO/IEC TR 15446: Information technology—Security techniques—Guide for the production of Protection Profiles and Security Targets (2009)
11. Bundesamt für Sicherheit in der Informationstechnik. Guidelines for Developer Documentation according to Common Criteria, Version 3.1 (2007)
12. CC Toolbox. http://niatec.info/ViewPage.aspx?id=44. Accessed 10 Mar 2017
13. Horie, D., Yajima, K., Azimah, N., Goto, Y., Cheng, J.: GEST: a generator of ISO/IEC 15408 security target templates. In: Lee, R., Hu, G., Miao, H. (eds.) Computer and Information Science 2009. SCI, vol. 208, pp. 149–158. Springer, Heidelberg (2009). http://link.springer.com/chapter/10.1007%2F978-3-642-01209-9_14#page-1. Accessed 10 Mar 2017
14. TL SET. http://trusted-labs.com/security-consulting/tools-training/tl-set/. Accessed 10 Mar 2017
15. CCMODE: Common Criteria compliant, Modular, Open IT security Development Environment’. http://www.commoncriteria.pl/. Accessed 10 Mar 2017
16. Goertzel, K.M., Winograd, T.: (Contributor): Information Assurance Tools Report – Vulnerability Assessment. 6th edn. Information Assurance Technology Analysis Center (IATAC), USA (2011)
17. ISO 31000:2009, Risk management – Principles and guidelines
18. ISO/IEC 31010:2009 – Risk Management—Risk Assessment Techniques

19. Bagiński, J., Rogowski, D.: Software support for enhanced risk management. In: Rostański, M., Pikiewicz, P., Buchwald, P., Maczka, K. (eds.): Proceedings of the XI International Scientific Conference Internet in the Information Society, Publishing University of Dąbrowa Górnicza, Cieszyn, Poland, 22–23 September 2016, pp. 369–388 (2016)
20. ISO/IEC TS 30104 Information technology—Security Techniques—Physical Security Attacks, Mitigation Techniques and Security Requirements (2015)

On the Performance of Some C# Constructions

Ilona Bluemke(✉), Piotr Gawkowski, Waldemar Grabski,
and Konrad Grochowski

Institute of Computer Science, Nowowiejska 15/19, 00-665 Warsaw, Poland
{I.Bluemke,P.Gawkowski,W.Grabski,
K.Grochowski}@ii.pw.edu.pl

Abstract. In some types of dependable applications (e.g. controlling some electronic devices) the execution time of a program has to be very short to enable the appropriate control of the device. Implementing code for Kamika's device that measures small particles in the air or in the liquid we are using C# language. Some parts of the code were also transformed into C++ to find overheads. The main part of the paper are results of some comparative experiments measuring the performance of alternative C# constructions. We were especially interested in extension methods which enable to "add" methods to existing types without creating a new derived type, recompiling, or modifying the original type.

Keywords: C++ · C# · Performance · Extension methods

1 Introduction

At the Institute of Computer Science, a software for the three-dimensional analyzer (called IPS) of micron-size particles being designed and produced by KAMIKA Instruments [10] is under development. This project is founded by the National Research and Development Centre (NCBiR in Polish). KAMIKA Instruments is Polish producer with rich experience in laboratory and industrial sector. Kamika has been producing measuring devices for thirty years and offers optical-electronic instruments for measurement of: materials' granulation in the air or in the liquid, pollution, drops of liquid and aerosol grain. Kamika's measuring system can be used e.g. for: control and optimization of industrial processes, research, measure the cleanness of water and air.

The IPS device consists of an optical head and a fast A/D converter (ADC) card. The substance particles are sucked by the air into the measuring pipe with a compressor. The feeder is responsible for accurate control of the rate at which particles flow through the pipe. In the measurement process two optical, crossed channels are used in the measuring pipe. Each channel uses a light source and a sensor. The way the optical measuring channels are made is a trade secret of Kamika. The main idea of the measurement process is to measure how the particle will shade the light as it flies between the light source and the sensor. As the channels are crossed, the amplitudes of impulses for each channel are correlated to one dimension (width - X for channel 1 and depth – Y for channel 2) of the particle. The width of the impulses is related to the height (Z) while being affected by the speed of the particle within the measuring pipe

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_4

(related to the compressor speed). As the sizes of the measured particles can be even single micrometers, a lot of optical effects might affect the measurement process, so high precision manufacturing is required. Also thorough calibration is obligatory. The light level from the sensors (in each channel) are sampled by the A/D converter. In two-channel configuration, the sampled values are gathered by the ADC card with the sample rate of 6 mega samples, i.e. 3 000 000 samples per second from each channel. Each sample is represented as a short (16 bits) value within the range of 0÷4095. As a consequence, the IPS collects 3 000 000 * 2 * 2 = 12 megabytes each second.

The IPS uses USB 2.0 interface to communicate with the workstation (PC). A special driver for the IPS is provided as well as dynamically loaded library (DLL) that implements basic programming interface (`startAcquisition`, `receiveBuffer`, `stopAcquisition`, etc.). The collected data can be read by the application in portions – fixed size buffers. The data is interleaved, i.e. odd short values in a buffer are samples from channel 1 while even values are channel 2 samples. The typical size of a buffer is 512 kilobytes (256 kilo-samples from both channels).

The ADC card has very limited resources. Each buffer has to be read in time, which is less than the time to collect data in the second buffer. Otherwise, the data will be overwritten. For the 512 kB buffer this is approximately only 43 ms and such value is very severe performance constraint. In the first step of processing, the data for each channel has to be decoupled. Then, in each channel, the stream processing includes averaging, zero-level tracking, detection of impulses, analysis of impulses against coincident (several particles are coupled or are too close to each other), and merging impulses from both channels. The statistic about each buffer (e.g. number of qualified particles, coincidental events) is necessary to control properly the feeding of the substance being measured. As a result, the analysis of the data has to be made in real time – not delayed for offline processing, etc. So, it is crucial to assure the high efficiency of processing while keeping the overall system software architecture simple for further development.

In the software we are designing and developing for Kamika, the C++ [1] and C# [2] programming languages are used. C++ produces native code which has to be compiled, with some optimization options, to run on a particular machine, while C# produces code in intermediate language (named as IL) which runs on a virtual machine. The JIT compiler, compiles IL the first time it's executed, and can make optimizations that a C++ compiled program cannot because it can query the machine and can adapt the code to the available instruction set or hardware features. Another factor is that JIT compiler cannot spend too much time on advanced optimizations, because it runs at runtime, and the end user would notice if it takes too much time. On the other hand, a C ++ compiler has all the time it needs to do optimizations at compile time. Therefore, it is not quite clear which code in C++ or in C# will perform better.

We were looking in the literature for comparisons of C++ and C# performances. The following data bases were searched: Springer Link, IEEE/IEE Electronic Library, Elsevier, ACM Digital Library, and Wiley Online Library. Unfortunately, only two works were found, i.e. [12, 14]. More information was available on Internet pages. Experiments, described in Sect. 2, show that the performance of C++ and C# programs depend on the language constructs used in the program code, algorithms, and environments. Some experiments show predominance of C++ while other of C#. In these experiments we were not able to find language constructs used in our code, so we conducted our own experiments and their results are presented in Sect. 3. Section 4 contains some conclusions.

2 Related Work

In the Internet there are several comparisons of the effectiveness of programs written in C++ and C# e.g. [3–10]. Some authors find the advantage of C++ over C# while others claim that C# can execute faster than C++.

C++ code is optimized by the compiler in the same way for all computers while C# code may be optimized for special machine (e.g. AMD, Intel, Pentium).

C# has many features which are very useful while writing applications, e.g. many libraries. If the programmer is not able to find a function in the standard library, maybe can find a C or C++ library for it, and quickly create a wrapper class for a C#. C# seems to be better suited to applications that need to be developed quickly and to complex projects where organization is important.

In [3] the author claims, that the main problem with C# compared to C++ is high memory consumption, which eventually slows down .NET software. In [3] the overhead of virtual method calls and object creation is also discussed. Virtual methods prevent optimizations such as method call in-lining. C++ has an advantage because templates may be used to achieve a different kind of generalization with no impact on runtime as they are resolved at compile time. In such cases C++ wins against C#.

In [6] the author made a fair comparison between the performance of C# and C++ code in desktop and mobile (Windows CE) environments. The majority of the code was written in C# first, and manually, line-by-line, translated to C++. Standard .NET classes were replaced by STL equivalents. Also cross-language benchmarks written by two different people and a variation of hash table benchmark using sorted maps were added. The main goal was to measure the performance difference in the compiler/JIT and parts of the standard library that are often used. The tests were limited to C++ standard library. The benchmarks were limited to the following areas*: string handling, hashtables, binary trees, simple structures, mathematical generics, simple arithmetic (add, subtract; multiply, divide, modulo) for different data types, 64-bit integers, text file scanning (read text files line by line), sorting* [6]. The author claims that for desktop application a programmer, which is equally skilled in C# and C++, can advantage of the simpler syntax, rich standard libraries, stellar IntelliSense, and reduced development effort that C# offers. The detailed results showed only a small performance penalty for C# versus C++.

In [7] other comparison of performance C# versus C++ is briefly described. The subject of the experiment was the performance of *binary search*. The test started with an "overhead" run, where a dummy search routine was called to assess the calling overhead. Next, an "iterative" algorithm and a "recursive" algorithm (calling itself in order to search the appropriate segment of the input array) were executed. The results using an array of 1 million values and doing 10 million searches, under C# and C++ are following: C# - overhead 0.0676 s, iterative 1.4688 s, recursive 2.5938 s; C++ - overhead 0.4720 s, iterative 2.0400 s, recursive 6.8290 s. The results are rather astonishing.

Unfortunately, the author hasn't described the environment in which this experiment was made nor explained the shorter times of C# programs.

Josiah Manson conducted also an experiment comparing the performance of C++ and C# programs and described it in [9]. As he is interested in programming games, he coded a ray tracer in C#. He did not optimized the code and used integers and doubles to store numbers. He was using mostly triangles in scenes, which were treated generically by inheriting attributes from a base shape class. All pixel colors were treated similarly, through generic shaders that inherit from a base class. When the code was bug-free he ported the code to C++ as identically as possible for a fair comparison. He traced different types of scenes that have varied code paths, amount of calculation, and memory access patterns. All the testing was on Windows 7 64-bit. Visual Studio 2010 Beta 2 for both C# and C++ was used, C++ was compiled with optimization options. For each scene initialization time and actual ray tracing time were measured separately, startup and shutdown time for the program were ignored. All times were measured in seconds. Manson concludes that C# is slow, approximately five times slower than identical C++ code on average.

Florent Clairambault in [8] also describes an experiment - comparison of C++ and C# (also Java) performance conducted on nine algorithms. He showed that C# can actually be considered as a very fast language. For four algorithms used in this experiment (i.e. Binary Tree, Spectral Norm, n-Sieve bits, and n-Sieve) the execution times were shorter for C# than C++ in Windows environment. In other algorithms (Spectral Norm, N Body, Mandelbrot, Fankuch, Fasta, and Recursives) C++ appeared to be faster. Florent Clairambault explains the good results of the C# with efficient hardware-specific optimization available in C# and not possible in C++.

In [12] Gerlach and Kneis compare the performance of the Bellman-Ford single source shortest path algorithm [13] when using parameterized data structures implemented in C++, Java, and C#. Theirs measurements showed that both Java and C# generics introduce only little run time overhead when compared with non-parameterized implementations. A C++ implementation that heavily was using templates has served as a reference implementation.

In [14] Gherardi, Brugali, and Comotti present their work on quantifying the difference of performance between Java and C++. They measured the execution times of an algorithm written in Java and in C++. The chosen algorithm was the Delaunay triangulation and its implementation was from the OSG library [15]. The algorithm is well suited for the purpose of comparison of performance because it stresses several critical points of the programming languages performance such as the frequent access to the memory for operating on dynamic size array and the frequent evaluation of logical conditions. Results showed that, using the appropriate optimizations, Java is from 1.09 to 1.51 times slower than C++ under Windows and from 1.21 to 1.91 times under Linux.

3 Experiments

During the measurement process (briefly described in Sect. 1) software must read data from Kamika' electronic device. Currently 12 megabytes appear in one second, split into samples. Each sample is 16 bits number which is written to a buffer. Buffer filled with samples can be read by USB interface. The producer of the electronic device provides a driver and a library (64 bits) for Windows system. With functions from this library we can read a buffer, establish its length, etc. Our application reads periodically the buffer (12 million of bytes per second) and has to read and process the buffer before the next buffer is filled. The device uses two buffers, while one is being filled the second one is available for reading. Our goal was to quantify the difference between the execution times of several C# and C++ constructs and to choose the constructs which offer good code readability and enable the processing of samples. As in the literature (see Sect. 2) we were not able to find information concerning the constructs suitable for our software, we carried out many experiments, some of them are described below.

3.1 Experiment with Reading Buffer

As the results of experiments described in Sect. 2 are not clear and we were not able to find the answer if C# application will manage to read the buffer while the other one is being filled we conducted our own experiments. Two simple application were written:

- one in C++,
- second in C# (Microsoft.Net).

Application periodically reads buffers with data and writes them to a file. Before two successive readings, a delay (slightly less than the period the buffer is enabled for reading) is introduced (function `Sleep`). The simple test application was reading 1200 buffers, checking if the data were not lost. The time of reading 1200 buffers and the processor load were gathered. It is worth to note, that the C# application has to use data marshalling in order to read the buffer from the device library. The experiment was made on a computer with: 2x Dual Core AMD Opteron 280 2.40 GHz processor, 8 GB memory and operating system Windows 7 Professional x64.

Two buffer lengths were examined (1048576 and 524288) and the delays were respectively set to 60 and 30 ms. The detailed results are given in Table 1. It appeared that the processor usage is similar for C++ and C# application, so, we decided to use C# as the implementation language for the component reading measured raw data. For smaller buffer lengths than these shown in Table 1, C# application was "loosing" some data (buffers were overwritten with new values) while C++ application not.

3.2 Comparison of Some Language Constructs

The comparison of selected language constructs were conducted on processor Intel Core I7 720QM 1.6 GHz, with 8 GB memory and operating system Windows 10 Enterprise x64. We were using prototype module `NoiseAnalysisApp` – a part of

Table 1. Results of reading buffer test

Buffer size	Lang.	Application latency	Reading time for 1200 buffers	Attempts to read	Processor performance [%]		
					min.	avg.	max.
1048576	C++	60	105118	1679	0	0.01	1.5
1048576	C#	60	105156	1679	0	0.057	1.562
524288	C++	30	52715	1684	0	0	0
524288	C#	30	52687	1681	0	0	0

the system being developed for Kamika. In this program one thread is reading data from a file while other threads are responsible for noise analysis in each channel stream (maximum two). In the "analyzing" thread the histogram of noise and its mode are produced. The time to calculate these values and the number of samples processed in a second are measured. As test data, we used four files given by Kamika with noise for one- and two-channel measuring device. These files contained approximately 400 buffers of the size of 524 288 samples (approx. 200 000 000 samples in each file).

We conducted several experiments checking how different C# language constructs and the organization of processing the stream of data influence the performance of processing samples. The application for Kamika measuring device has to be able to process 12 million of samples in a second. In each of these experiments we measured the execution time of `MakeNoiseAnalysis` method. In all experiments the calculations were similar – we only changed the way samples were extracted from the buffer and the calculation of moving average values. Each experiment was repeated three times and the mean values were calculated. First we prepared a "reference version" – with classic *for*-loop, without iterators and extension methods. The code is shown in Fig. 1

Experiments

We were examining different ways of reading samples from buffers and calculating moving average. The detailed results for two stream measuring device and for the following list of experiments are shown in Table 2:

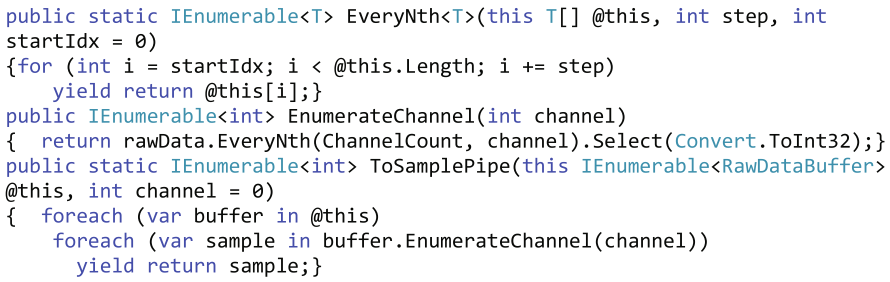
```
voidMakeClassicNoiseAnalysis(IEnumerable<RawDataBuffer>bufferSource)
{
DateTimestartTime = DateTime.Now;
varaverager = newMovingAverage(15);
foreach (var buffer inbufferSource)
{
for (int i = Channel; i <buffer.rawData.Length;i+=buffer.ChannelCount)
{
var sample= buffer.rawData[i];
varavg_sample = averager.ProcessNextSample(sample);
ProcessSample(avg_sample);
}
}
Result.ProcessingTime = DateTime.Now - startTime;
}
```

Fig. 1. Reference version of `MakeNoiseAnalysis` method

Table 2. Results of experiments

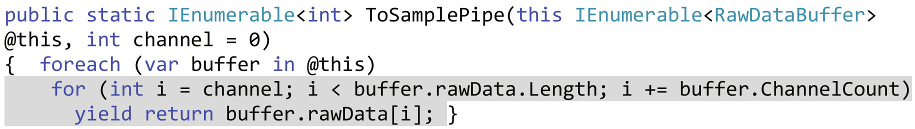
Line	Experiment	No of samples	Samples per sec	Processing (ns)	% Change to the basic version
1	reference	178 782 208	25 911 417	77	
2	reference	208 666 624	26 366 040	76	
3	1	178 782 208	25 420 771	79	98
4	1	208 666 624	25 324 939	79	96
5	2	178 782 208	23 272 527	86	90
6	2	208 666 624	23 165 688	86	88
7	3 –**Fig. 2**	178 782 208	15 603 334	128	60
8	3 –**Fig. 2**	208 666 624	15 618 337	128	59
9	4 –**Fig. 3**	178 782 208	19 870 138	101	77
10	4 –**Fig. 3**	208 666 624	20 118 353	99	76
11	5 –**Fig. 4**	178 782 208	9 891 741	202	38
12	5 –**Fig. 4**	208 666 624	9 905 229	202	38
13	6 – **Fig. 5**	178 782 208	10 333 851	194	40
14	6 – **Fig. 5**	208 666 624	10 236 052	195	39

1. We change the way buffers with samples are read. Instead of interface `IEnumerable` (line 1 in Fig. 1) and `foreach` loop we created our own interface. The execution time was similar, so, we can reason that constructs `IEnumerable` and `foreach` are not decreasing the performance (lines 3-4, Table 2).
2. We added a class – classic iterator, providing successive samples from buffers. The decrease in performance was 10–13% only. We observed that iterators make the source code more easy to read and understand but slightly decrease its performance (lines 5–6, Table 2).
3. We prepared extension method `ToSamplePipe`, which internally uses extension method `EnumerateChannel` and this one is using extension method `EveryNth` for taking samples from appropriate channel. Those methods build up an `IEnumerable` stack, similar to LINQ [11] library. The code of these methods is shown in Fig. 2. In the `MakeNoiseAnalysis` method line 7 (Fig. 1) was substituted with: `foreach (var sample in bufferSource.ToSamplePipe())`. The results showed significant performance decrease – about 40% (lines 7–8, Table 2).
4. Next, instead of extension method `EnumerateChannel`, for reading successive samples we used `for` and `yield` instructions (Fig. 3). The performance was 75–78% of the performance of the first approach (Fig. 1) but was over a dozen greater than in point 3. This experiment revealed that the usage of extension methods significantly slows program execution (lines 9–10, Table 2).
5. The extension method `ToSamplePipe` was used for reading samples and extension method `MovingAverage` was calculating the moving average (Fig. 4). In the method `MakeNoiseAnalysis` we substituted some instructions



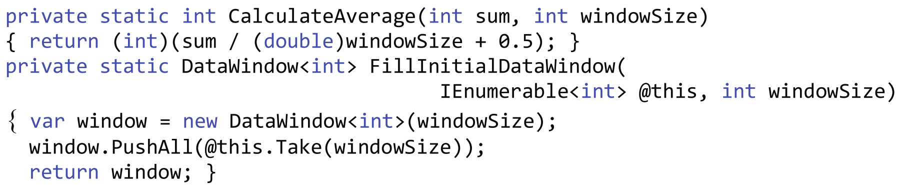
```
public static IEnumerable<T> EveryNth<T>(this T[] @this, int step, int
startIdx = 0)
{for (int i = startIdx; i < @this.Length; i += step)
    yield return @this[i];}
public IEnumerable<int> EnumerateChannel(int channel)
{  return rawData.EveryNth(ChannelCount, channel).Select(Convert.ToInt32);}
public static IEnumerable<int> ToSamplePipe(this IEnumerable<RawDataBuffer>
@this, int channel = 0)
{  foreach (var buffer in @this)
    foreach (var sample in buffer.EnumerateChannel(channel))
      yield return sample;}
```

Fig. 2. Extension methods used for reading samples



```
public static IEnumerable<int> ToSamplePipe(this IEnumerable<RawDataBuffer>
@this, int channel = 0)
{  foreach (var buffer in @this)
    for (int i = channel; i < buffer.rawData.Length; i += buffer.ChannelCount)
      yield return buffer.rawData[i]; }
```

Fig. 3. Loop and yield instructions used for reading samples

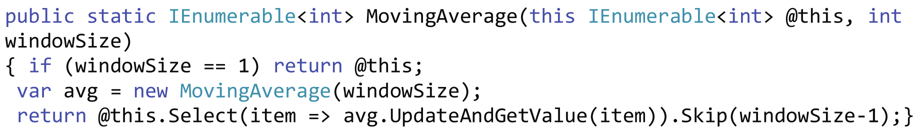


```
private static int CalculateAverage(int sum, int windowSize)
{ return (int)(sum / (double)windowSize + 0.5); }
private static DataWindow<int> FillInitialDataWindow(
                                     IEnumerable<int> @this, int windowSize)
{ var window = new DataWindow<int>(windowSize);
  window.PushAll(@this.Take(windowSize));
  return window; }
```

Fig. 4. Extension method `MovingAverage`

with: `foreach (var sample in bufferSource.ToSamplePipe().MovingAverage(15))`. The code with these methods looks very elegant, is easy to understand but the performance of such code was unacceptable, only 40% of the performance from the basic version was achieved. In our program such slow execution disabled appropriate communication with the measuring device (lines 11–12, Table 2).

6. In this experiment we changed extension method `MovingAverage`, which is using basic class `MovingAverage` to calculate the average moving value and extension methods *Skip* and *Take* of interface `IEnumerable`. The code is shown in Fig. 5. The performance is similar to the performance from the previous point (lines 13–14, Table 2).
7. The cost of an LINQ-style `IEnumerable` stack was also examined. Previous experiments shown that such coding style significantly slows down the execution of a program. We wanted to have the detailed value, how much time each stack element adds to the time of processing sample. We measured that each layer adds 28–35 ns to the processing of each sample. This is quite a significant value because it is approximately 25% of the time of processing a sample without stacking `IEnumerable`.



```
public static IEnumerable<int> MovingAverage(this IEnumerable<int> @this, int
windowSize)
{ if (windowSize == 1) return @this;
 var avg = new MovingAverage(windowSize);
 return @this.Select(item => avg.UpdateAndGetValue(item)).Skip(windowSize-1);}
```

Fig. 5. Other implementation of method *MovingAverage*

8. We also examined the cost of classic calls of methods. The cost appeared to be insignificant. For single call is was unmeasurable, for several calls was 30 times less than the cost of structures built with extension methods (experiment described in point 7).

3.3 Summary

We examined the performance of several C# constructs. It appeared that interface `IEnumerable` and `foreach` loop do not decrease significantly the performance of the code. Stacking `IEnumerable` in a similar way as LINQ library does (using extension methods, e.g. `ToSamplePipe`, `MovingAverage`, `AddToSample` – Sect. 3.2) is “expensive” in terms of the performance. The time of the call of `IEnumerable` wrapping another `IEnumerable` is 30 times longer than the regular call. The usage of LINQ-style extension methods makes the source code “elegant”, more “functional programing style”, easy to use but significantly slows down the execution of a program. In our application, which should process about 12 million samples in one second, extensions methods can’t be widely used. It appeared that classic calls of methods are very effective and should not be avoided.

The reason behind performance drop hides in the cost of the virtual function call – each `IEnumerable` accesses the next in layer via `MoveNext` and `Current` virtual calls, even if those calls are not directly used by the programmer (`yield` construct can shadow them). Our extension method was in reality creating another class implementing `IEnumerable` interface and, as such, providing the next level of virtual calls, which C# compiler and JIT failed to optimize. LINQ library uses several tricks to overcome those bottleneck (`Select(…).Where(…)` does introduce only single layer), but the result is not as readable as extension methods with yields.

4 Conclusions

The results of the presented experiments proved that application written in C# (environment Microsoft.Net) may be used to process in real time the data stream obtained from a measuring device, even quite fast. We also shown that some programming constructs, nice and useful for a programmer, may significantly decrease the performance of an application. Extension methods and LINQ-style expressions make the source code clear, are easy to use but are significantly (about 30 times) slower than the classic ones – so, can’t be used in applications for which the performance is significant.

Classic methods call and *for*-loops or inline functions are quite efficient and should be used. It is worth mentioning that a programmer who knows the programming language very well will certainly produce faster code than the one who doesn't, regardless of the languages. Algorithm choice and implementation have a more intense effect on performance than implementation language chosen. Programmers should concentrate on first implementing the application correctly in whatever language suits them best, then finding performance bottlenecks, and then optimize the code. Sometimes critical parts of a system can be written in lower level language. The performance depends also on the software architecture. In [16] software performance antipatterns that represent "bad practices" decreasing performance are introduced.

Acknowledgments. The research leading to the presented results has received funding from the National Centre for Research and Development grant number POIR.01.01.01-00-0459/15-00.

References

1. Stroustrup. B.: The C++ Programming Language, 4th edn. Addison Wesley, Reading (2013). ISBN 0-321-56384-0
2. Troelsen, A., Japikse, P.: C# 6.0 and the .NET 4.6 Framework, 7th edn. Apress (2015). ISBN: 978-1-4842-1333-9, doi:10.1007/978-1-4842-1332-2
3. http://stackoverflow.com/questions/138361/how-much-faster-is-c-than-c (Accessed Jan 2017)
4. http://stackoverflow.com/questions/145110/c-performance-vs-java-c (Accessed Jan 2017)
5. http://www.cplusplus.com/forum/general/75347/ (Accessed Jan 2017)
6. http://www.codeproject.com/Articles/212856/Head-to-head-benchmark-Csharp-vs-NET (Accessed Jan 2017)
7. https://social.msdn.microsoft.com/Forums/en-US/37db0dc6-aee5-4947-ba02-7dc63a87e09d/c-vs-c-performance-test?forum=csharplanguage (Accessed Jan 2017)
8. http://florent.clairambault.fr/stupid-cpp-vs-charp-performance-comparison (Accessed Jan 2017)
9. http://josiahmanson.com/prose/speed_cpp_csharp/ (Accessed Jan 2017)
10. Kamika.pl/en (Accessed Jan 2017)
11. https://msdn.microsoft.com/en-us/library/bb397933.aspx (Accessed Jan 2017)
12. Gerlach, J., Kneis, J.: Generic programming for scientific computing in C++, Java, and C#. In: Zhou, X., et al. (eds.) APPT 2003. LNCS, vol. 2834, pp. 301–310. Springer (2003)
13. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. MIT Press (1998)
14. Gherardi, L., et al..: A Java vs. C++ performance evaluation: a 3D modeling benchmark. In: Noda, I., et al. (eds.) SIMPAR 2012. LNAI, vol. 7628, pp. 161–172. Springer (2012)
15. Open Scene Graph. http://www.openscenegraph.org (Accessed 2012)
16. Cortellessa, V., et al.: Software performance antipatterns: modeling and analysis. In: Bernardo, M., et al. (eds.) SFM 2012. LNCS, vol. 7320, pp. 290–335. Springer (2012)

Deep Stacking Convex Neuro-Fuzzy System and Its On-line Learning

Yevgeniy Bodyanskiy[1], Olena Vynokurova[1(✉)], Iryna Pliss[1],
Dmytro Peleshko[2], and Yuriy Rashkevych[2]

[1] Kharkiv National University of Radio Electronics,
14 Nauky av., Kharkiv 61166, Ukraine
{yevgeniy.bodyanskiy, iryna.pliss}@nure.ua,
vynokurova@gmail.com
[2] Lviv Polytechnik National University, 12 Bandera Street, Lviv 79013, Ukraine
dpeleshko@gmail.com, rashkev@polynet.lviv.ua

Abstract. In the paper the architecture of Deep Stacking Convex Neuro-Fuzzy System for data stream processing in on-line mode is proposed. The advantage of proposed system is that its layers are formed by multivariate modification of hybrid generalized additive neuro-fuzzy system. Such system is characterized by simplicity of computational implementation, high speed learning, increased approximation properties. For learning of the proposed system both conventional least squares method (including its recurrent version) and specialized learning procedures, which have tracking and smoothing properties are used. Proposed system is aimed at solving of wide range of Data Stream Mining problems, which are connected with processing of nonstationary stochastic and chaotic processes under conditions when information is fed to the system in on-line mode.

Keywords: Computational intelligence · Deep learning · Deep stacking convex neuro-fuzzy system · Hybrid systems

1 Introduction

Nowadays hybrid systems of computational intelligence, which are formed based on conventional artificial neural networks, are wide spread for a lot of information processing tasks, including tasks of Data Stream Mining where data are fed for processing in on-line mode in the form of multivariate time series. It is clear that not all neural networks and, first of all, conventional multilayer perceptrons, and also their hybrids (neuro-fuzzy systems) can operate in this mode. That is why, such systems demand significant improving and modification for possibility to operate under conditions when data are fed to the processing in sequential mode, one after another observation. Recent years, the information processing systems based on deep learning (first of all, deep neural networks [1–4]) call attention to itself. Such systems demonstrate high quality of obtained results, however at that demand high time spending for learning in comparison with conventional shallow neural networks. The learning time can be reduced using so-called deep stacking convex neural networks [2], whose layers are formed by

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_5

simplified neural networks modules, and learning problem is reduced to convex optimization, which has an analytical solution. It is clear that having analytical solution, it is easy to realize recurrent on-line learning process. Generally implementing of on-line learning process is possible for neural networks, whose output signal depends linearly from tuned synaptic weights, for example, Radial Basis Function Networks (RBFN) [5–7], and Normalized Radial Basis Function Networks (NRBFN) [8, 9], however their using is often complicated by, so called, the curse of dimensionality. In addition, here the key moment is not computational complexity, but the problem is obtaining of data sets from the real plant that can be too small for estimating of large synaptic weights number. Neuro-fuzzy systems that combine the learning ability of neural networks and transparency and interpretability of the soft computing methods, have a range of advantages ahead of the conventional neural networks. Here, first of all, it should be noticed TSK-system [10], and ANFIS [11], whose output signal also depends linearly from the synaptic weights and has less number of synaptic weights than RBFN or NRBFN. The more complex hybrid systems of computational intelligence are well-known and have improved approximation properties, for example, the hybrid fuzzy wavelet neural networks [12, 13], but learning algorithms complexity limits their using in on-line mode. Therefore in deep stacking convex systems of computational intelligence it is appropriate to use multivariate adaptive hybrid neuro-fuzzy systems that allow to process nonstationary information that is disturbed by noises in on-line mode, have smaller number of tuned parameters comparatively with known neuro-fuzzy systems, is simple in the computational implementation (due to the paralleling of the information processing) and doesn't demand previous defining of the training set, i.e. to implements the learning process starting with the first observation, which is fed to the system.

2 Architecture of Deep Stacking Convex Neuro-Fuzzy System

Figure 1 shows the architecture of proposed deep stacking convex neuro-fuzzy system, which is hybrid of cascade-correlation neural network of Fahlman-Lebiere [14] and cascade neuro-fuzzy systems [15–17] with a priori unknown number of layers that can increase steadily. The proposed system realizes a nonlinear mapping $R^n \to R^m$, that is provided by sequentially connected layers of information processing, which number g is defined directly under learning process. On the input of the first hidden layer vector $x(k) = (x_1(k), \ldots, x_n(k))^T \in R^n$ is fed and the output of this layer is vector $\hat{y}^{[1]}(k) = (\hat{y}_1^{[1]}(k), \ldots, \hat{y}_m^{[1]}(k))^T \in R^m$; the input of the second layer is signal $(x^T(k), \hat{y}^{[1]T}(k))^T \in R^{n+m}$ and its output - $\hat{y}^{[2]T}(k) \in R^m$; the input and output of the third layer are $(x^T(k), \hat{y}^{[1]T}(k), \hat{y}^{[2]T}(k))^T \in R^{n+2m}$,$\hat{y}^{[3]}(k) \in R^m$, and finally, the input of g-th layer is $(x^T(k), \hat{y}^{[1]T}(k), \ldots, \hat{y}^{[g-1]T}(k))^T \in R^{n+(g-1)m}$, and output is $\hat{y}^{[g]T}(k) \in R^m$. This system architecture allows not to use for learning the conventional errors back propagation algorithm, but to tune each layer sequentially using output signals from previous layers. At that if output signal of each layer depends linearly from the tuning parameters, the learning process is transformed into convex optimization.

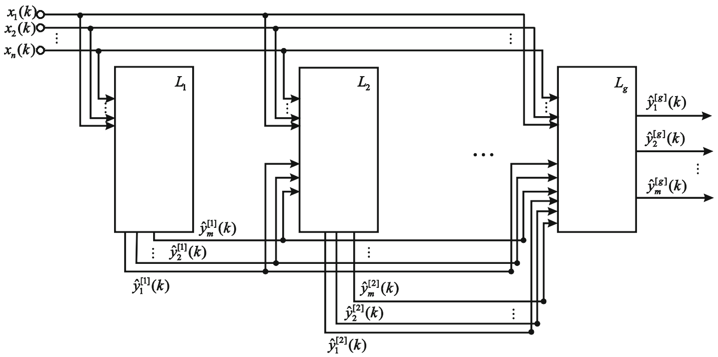


Fig. 1. Deep stacking convex neuro-fuzzy system

3 The Zero-Order Takagi-Sugeno-Kang Neuro-Fuzzy System

The zero-order Takagi–Sugeno–Kang (TSK)-type neuro-fuzzy system (known as Wang-Mendel neuro-fuzzy system too) [18] is one of the most effective ones for processing of information that is fed in real time. Such system is characterized by relative small number of tuned synaptic weights that can be adjusted by high-speed second order optimization procedures such as recurrent least squares learning algorithm.

The architecture of neuro-fuzzy system by Wang-Mendel consists of five sequentially connected layers. To the input layer $(n \times 1)$-dimensional vector of the input signals $x(k) = (x_1(k), \ldots, x_i(k), \ldots, x_n(k))^T$ is fed that has to be processed, where $k = 1, 2, 3, \ldots$ is current instant time. The first hidden layer consists of nh membership functions $\mu_{li}(x_i)$, $l = 1, 2, \ldots, h$ and provides fuzzyfication of the input variables. The second layer implements the aggregation of membership levels that are computed in the first layer and consists of h multiplier units. The third hidden layer consists of synaptic weights that have to be adjusted during learning process of neuro-fuzzy system. The forth layer contains usual summators and computes sums of the output signals of the second and third layers. And, finally, the fifth (output) layer produces a normalization where as the result output signal $\hat{y}_p(k)$ $(p = 1, \ldots m)$ is computed.

Therefore this system implements the nonlinear mapping $x \in R^n \Rightarrow \hat{y} \in R^m$. If the vector signal $x(k)$ is fed to the system input than the first layer elements compute the membership levels $0 < \mu_{li}(x_i(k)) \le 1$, at that as membership functions the bell-shaped structures with nonstrictly local receptive field are used, that allows to avoid "gaps" appearance in the fuzzificated space under using scatter partition of the input variables space. The most often as a membership functions of the first layer the Gaussian functions are used in the form

$$\mu_{li}(x_i(k)) = \exp\left(-(x_i(k) - c_{li})^2 \Big/ 2\sigma_i^2\right) \tag{1}$$

where c_{li}, σ_i are center and width parameters respectively. These parameters can be chosen by empirical way or tuned during learning process by the backpropagation learning algorithm. It can be also noticed that previous input variables coding on the limited interval, for example, $0 \le x_i(k) \le 1$, allows to simplify the computations because the width parameter σ_i can be set equal for all components of the input vector.

The second hidden layer outputs compute aggregated values in the form

$$\tilde{x}_l(k) = \prod\nolimits_{i=1}^{n} \mu_{li}(x_i(k)) \tag{2}$$

and for Gaussian functions with equal values of width parameters σ we can write

$$\tilde{x}_l(k) = \prod\nolimits_{i=1}^{n} \exp\left(-(x_i(k) - c_{li})^2 \Big/ 2\sigma^2\right) = \exp\left(-\|x(k) - c_l\|^2 \Big/ 2\sigma^2\right) \tag{3}$$

(here $c_l = (c_{l1}, \ldots, c_{li}, \ldots c_{ln})^T$), i.e. elements of the first and second hidden layers process the input signal similarly radial basis neurons in RBFN and NRBFN.

The outputs of the third hidden layer are values in form

$$w_{pl}(k-1) \prod\nolimits_{i=1}^{n} \mu_{li}(x_i(k)) = w_{pl}(k-1)\tilde{x}_l(k);\ p = 1, 2, \ldots, m;\ l = 1, 2, \ldots, h \tag{4}$$

(here $w_{pl}(k-1)$ are mh values of synaptic weights that are computed using $k-1$ previous observations), the output of fourth layer is

$$\sum\nolimits_{l=1}^{h} w_{pl}(k-1) \prod\nolimits_{i=1}^{n} \mu_{li}(x_i(k)) = \sum\nolimits_{l=1}^{h} w_{pl}(k-1)\tilde{x}_l(k) \tag{5}$$

$$\sum\nolimits_{l=1}^{h} \prod\nolimits_{i=1}^{n} \mu_{li}(x_i(k)) = \sum\nolimits_{l=1}^{h} \tilde{x}_l(k) \tag{6}$$

and, finally, the system outputs (the fifth layer) produce signals in form

$$\hat{y}_p(k) = \frac{\sum_{l=1}^{h} w_{pl}(k-1) \prod_{i=1}^{n} \mu_{li}(x_i(k))}{\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k))} = w_p^T(k-1)\varphi(x(k)) \tag{7}$$

where $\varphi_l(x(k)) = \prod_{i=1}^{n} \mu_{li}(x_i(k)) \left(\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k))\right)^{-1}$, $w_p(k-1) = \big(w_{p1}(k-1),$ $\ldots, w_{ph}(k-1)\big)^T$, $\varphi(x(k)) = (\varphi_1(x(k)), \ldots, \varphi_h(x(k)))^T$.

It is simply to note that such system produces nonlinear mapping of the input signals into the output one similarly to normalized radial basis function network, however contains essentially smaller number of synaptic weights in comparison to NRBFN.

4 Multivariate Hybrid Neuro-Fuzzy System

Reducing of synaptic weights number in the neuro-fuzzy system by Wang-Mendel in comparison to normalized radial basis function network is achieved due to scatter partition of inputs, at that however in the areas, which are remoted from centers c_l of multidimensional membership functions in form

$$\prod_{i=1}^{n} \exp\left(-(x_i(k)-c_{li})^2 \Big/ 2\sigma^2\right) = \exp\left(-\|x(k)-c_l\|^2 \Big/ 2\sigma^2\right) \tag{8}$$

the provided quality of approximation can be nonsufficient.

Of course we can improve the approximation quality using grid partition of input space but at that the number of tuning parameters increases rapidly, i.e. the neuro-fuzzy systems advantages are lost ahead of the conventional neural network. For improving the approximation properties of neuro-fuzzy system we can introduce, so called, nonlinear synapses in the third hidden layer instead of usual synaptic weights w_{pl}, $p = 1, 2, \ldots, m,\ l = 1, 2, \ldots, h.$ These nonlinear synapses are building elements of neo-fuzzy neuron [19], which is enough simple and effective real-time system of computational intelligence, which is aimed at operating in on-board applications [20]. The neuro-fuzzy system based on neo-fuzzy neurons was proposed in [21] and its simplified version in [22]. This system confirmed their efficiency for many tasks connected with Dynamic Data Mining and Data Stream Mining. Here, it is necessary to notice that systems based on nonlinear synapses and neo-fuzzy neurons are the single output systems while for it is necessary to use multi inputs – multi outputs description a lot of real tasks' solution. In generally we can solve many tasks using some number of the parallel one-type single output systems. This approach was proposed in [23] where for solving of smart house tasks the group of parallel ANFIS have been used. At that, therefore, the implementation of such systems is getting more complicate and the number of tuning parameters is increased. In the connection with this, we have proposed [24] adaptive multivariate hybrid neuro-fuzzy system, which is characterized by the comparatively small number of adjustable parameters allows to tune parameters in real time under nonstationary and stochasticity of processed information conditions. Figure 2 shows the architecture of proposed multivariate hybrid neuro-fuzzy system, which is generalization of neuro-fuzzy system with single output for multivariate cases, which is considered in [25, 26]. The first and second layers are absolutely similar with the such layers of Wang-Mendel system, so we can write the values on their outputs in form

$$\tilde{x}_l(k) = \prod_{i=1}^{n} \mu_{li}(x_i(k)). \tag{9}$$

These signals are fed to the inputs of multivariate nonlinear synapses MNS_1, $MNS_2, \ldots, MNS_h$ that together with summator of fourth layer compose the generalized neo-fuzzy neuron architecture [27, 28].

Generalized neo-fuzzy neuron (GNFN) is nonlinear learning system with many inputs and outputs that implements mapping in the form

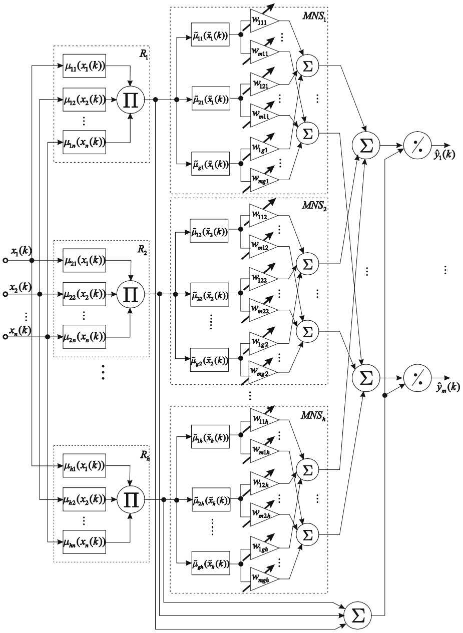


Fig. 2. Multivariate hybrid neuro-fuzzy system

$$f_p(\tilde{x}(k)) = \sum\nolimits_{l=1}^{h} f_{pl}(\tilde{x}_l(k)) \tag{10}$$

where $\tilde{x}(k) = (\tilde{x}_1(k), \ldots, \tilde{x}_l(k), \ldots, \tilde{x}_h(k))^T$, $f_p(\tilde{x}(k))$ is p-th output of GNFN ($p = 1, 2, \ldots, m$). Each nonlinear synapse MNS_l consists of g membership functions $\tilde{\mu}_{jl}(\tilde{x}_l)$, $l = 1, 2, \ldots, g$ and mg tuned synaptic weights w_{pjl}.

Therefore, the mapping, which is implemented by each multivariate nonlinear synapse, can be written in form

$$f_{pl}(\tilde{x}_l(k)) = \sum\nolimits_{j=1}^{g} w_{pjl}(k-1)\tilde{\mu}_{jl}(\tilde{x}_l(k)) \tag{11}$$

and generalized neo-fuzzy neuron realizes nonlinear mapping in form

$$f_p(\tilde{x}(k)) = \sum\nolimits_{l=1}^{h} \sum\nolimits_{j=1}^{g} w_{pjl}(k-1)\tilde{\mu}_{jl}(\tilde{x}_l(k)),\;\; p = 1, 2, \ldots, m \tag{12}$$

i.e. is actually multivariate generalized additive model [29, 30] that is characterized by computational simplicity and improved approximation properties.

The second integrator of the fourth hidden layer similarly to Wang-Mendel system computes the value

$$\sum\nolimits_{l=1}^{h}\prod\nolimits_{i=1}^{n}\mu_{jl}(x_i(k))=\sum\nolimits_{l=1}^{h}\tilde{x}_l(k) \tag{13}$$

and the outputs of system produce signals in form

$$\hat{y}_p(k)=\sum\nolimits_{l=1}^{h}\sum\nolimits_{j=1}^{g}w_{pjl}(k-1)\tilde{\mu}_{jl}(\tilde{x}_l(k))\Big/\sum\nolimits_{l=1}^{h}\tilde{x}_l(k)=w_p^T(k-1)\tilde{\varphi}(\tilde{x}(k)) \tag{14}$$

where

$$\tilde{\varphi}_{jl}(\tilde{x}(k))=\tilde{\mu}_{jl}(\tilde{x}_l(k))\left(\sum\nolimits_{l=1}^{h}\tilde{x}_l(k)\right)^{-1}=\tilde{\mu}_{jl}\left(\prod\nolimits_{i=1}^{n}\mu_{li}(x_i(k))\right)\left(\sum\nolimits_{l=1}^{h}\prod\nolimits_{i=1}^{n}\mu_{li}(x_i(k))\right)^{-1},$$

$$w_p(k-1)=(w_{p11}(k-1),w_{p21}(k-1),\ldots,w_{pg1}(k-1),w_{p12}(k-1),\ldots,w_{pjl}(k-1),\\ \ldots,w_{pgh}(k-1))^T,\ \tilde{\varphi}(\tilde{x}(k))=\left(\tilde{\varphi}_{11}(\tilde{x}(k)),\tilde{\varphi}_{21}(\tilde{x}(k)),\ldots,\tilde{\varphi}_{jl}(\tilde{x}(k)),\ldots,\tilde{\varphi}_{gh}(\tilde{x}(k))\right)^T.$$

It can be noticed that system under consideration is hybrid of the neuro-fuzzy system by Takagi-Sugeno-Kang and multivariate generalized additive model by Hastie-Tibshirani, which implements Takagi-Sugeno fuzzy inference and is protected from the problem which connects with the scatter partition. The output signals of such system depend linearly on the synaptic weights, whose number is in g times larger than in the system of Wang-Mendel and is the same as number of tuned synaptic weights of generalized neo-fuzzy neuron with h inputs. It is necessary to notice that as membership functions of nonlinear synapses MNS_l it can be used the same Gaussian functions, that are used in the first hidden layer. But if as $\tilde{\mu}_{jl}(\tilde{x}_l)$ we use membership functions, which satisfy to the conditions of unity partition (triangular, cubic and B-splines and etc.) the architecture of proposed system can be reduced by the rejection of one integrator in the fourth layer and devisors in the fifth (output) layer due to the fact that in this case we don't need to produce output signal normalization, i.e. in this case the output signal is computed similarly as in the standard neo-fuzzy neuron.

5 On-line Learning of the Proposed Deep Neuro-Fuzzy System

The learning process of the proposed system can be presented based on the first hidden layer L_1, at that such learning process is considered as process of synaptic weights estimation of GNFN in each layer of system. Let's introduce under consideration $(m\times 1)$ reference signals vector $y(k)=(y_1(k),\ldots,y_p(k),\ldots,y_m(k))^T$, $(m\times gh)$ matrix of synaptic weights in k-th instant of time $k=1,2,\ldots,N$

$$W(k)=\begin{pmatrix} w_{111}(k) & w_{121}(k) & \cdots & w_{1gh}(k)\\ w_{211}(k) & w_{222}(k) & \cdots & w_{2gh}(k)\\ \vdots & \vdots & \ddots & \vdots\\ w_{mn1}(k) & w_{m21}(k) & \cdots & w_{mgh}(k)\end{pmatrix}=\{w_{pjl}\} \tag{15}$$

and learning criterion

$$\begin{aligned} E(N) &= 1/2\sum\nolimits_{k=1}^{N} \|y(k) - \hat{y}(k)\|^2 \\ &= 1/2\sum\nolimits_{k=1}^{N}\sum\nolimits_{p=1}^{m} e_p^2(k) = 1/2\sum\nolimits_{k=1}^{N} \|y(k) - W\tilde{\varphi}(\tilde{x}(k))\|^2 \end{aligned} \tag{16}$$

Analytical minimization of this criterion leads to standard least squares estimate

$$W(N) = \sum\nolimits_{k=1}^{N} y(k)\tilde{\varphi}^T(\tilde{x}(k))\left(\sum\nolimits_{k=1}^{N} \tilde{\varphi}(\tilde{x}(k))\tilde{\varphi}^T(\tilde{x}(k))\right)^{-1}, \tag{17}$$

which for operation in on-line mode can be rewritten in the recurrent form

$$\begin{cases} W(k) = W(k-1) + \frac{(y(k)-W(k-1)\tilde{\varphi}(\tilde{x}(k)))\tilde{\varphi}^T(\tilde{x}(k))P(k-1)}{1+\tilde{\varphi}^T(\tilde{x}(k))P(k-1)\tilde{\varphi}(\tilde{x}(k))}, \\ P(k) = P(k-1) - \frac{P(k-1)\tilde{\varphi}(\tilde{x}(k))\tilde{\varphi}^T(\tilde{x}(k))P(k-1)}{1+\tilde{\varphi}^T(\tilde{x}(k))P(k-1)\tilde{\varphi}(\tilde{x}(k))}. \end{cases} \tag{18}$$

In the tasks of non-stationary data stream processing it is appropriate using of its exponentially-weighed modification instead of learning criterion (16)

$$E(k) = 1/2\sum\nolimits_{j=1}^{k} \alpha^{k-j}\|y(j) - \hat{y}(j)\|^2 = 1/2\sum\nolimits_{j=1}^{k} \alpha^{k-j}\|y(j) - W\tilde{\varphi}(\tilde{x}(j))\|^2 \tag{19}$$

(here $0 < \alpha \leq 1$ is forgetting factor), which minimization using Gaussian-Newtonian procedure leads to well-known algorithm that provides both tracking and high speed of convergence to optimal values of synaptic weights

$$\begin{cases} W(k) = W(k-1) + \frac{(y(k)-W(k-1)\tilde{\varphi}(\tilde{x}(k)))\tilde{\varphi}^T(\tilde{x}(k))P(k-1)}{\alpha+\tilde{\varphi}^T(\tilde{x}(k))P(k-1)\tilde{\varphi}(\tilde{x}(k))}, \\ P(k) = \frac{1}{\alpha}\left(P(k-1) - \frac{P(k-1)\tilde{\varphi}(\tilde{x}(k))\tilde{\varphi}^T(\tilde{x}(k))P(k-1)}{\alpha+\tilde{\varphi}^T(\tilde{x}(k))P(k-1)\tilde{\varphi}(\tilde{x}(k))}\right), \end{cases} \tag{20}$$

which can be unstable at small values of parameter α.

6 Experiment Results

Efficiency of hybrid neural network was examined based on solving forecasting problem of real ecological time series. This time sequence describes monthly pressure above the sea level from 1882 to 1998 (Darwin sea level pressure). This time series is a key indicator of climate change, as well as important in the study of the effect of the El Niño or Southern Oscillation index. Inputs number of network were taken as $n = 6$, that for input vector in the form $x(k-5), x(k-4), x(k-3), x(k-2), x(k-1), x(k)$ for the prediction value $x(k+1)$. As the quality criterion of forecasting root mean square error (MSE) was used (Table 1).

Thus as it can be seen from experimental results the proposed approach having the best quality of prediction in comparison with conventional neural network.

Table 1. Comparative analysis of forecasting time series results

Neural network/Learning algorithm	Number of layers	MSE
Deep stacking convex neuro-fuzzy system/Proposed learning algorithm	2	0,0135
Deep stacking convex neuro-fuzzy system/Proposed learning algorithm	4	0,0048
Hybrid multilayer GMDH-neural network based on wavelet neurons/RLSM	4	0,0398

7 Conclusions

Proposed deep staking convex neuro-fuzzy system is simple in computational implementation and has high speed of learning due to freedom from error backpropagation in the learning process. Such system can be used for solving wide spread of Data Mining tasks, including Data Stream Mining and Dynamic Data Mining tasks, and first of all, identification, forecasting, emulation, pattern recognition in on-line mode, for example, in on-board applications, which need the nonstationary data streams processing with high speed in sequential mode.

References

1. LeCun, Y., Bengio, Y., Hinton, G.E.: Deep learning. Nature **521**, 436–444 (2015)
2. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
3. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, New York (2016)
4. Peleshko, D., Ivanov, Y., Sharov, B., Izonin, I., Borzov, Y.: Design and implementation of visitors queue density analysis and registration method for retail videosurveillance purposes. In: 2016 IEEE First International Conference on Data Stream Mining and Processing, Lviv, Ukraine, pp. 159–162 (2016)
5. Rutkowski, L.: Computational Intelligence: Methods and Techniques. Springer, Berlin (2008)
6. Kruse, R., Borgelt, C., Klawonn, F., Moewes, C., Steinbrecher, M., Held, P.: Computational Intelligence. A Methodological Introduction. Springer, Berlin (2013)
7. Du, K.-L., Swamy, M.N.S.: Neural Networks and Statistical Learning. Springer, London (2014)
8. Nelles, O.: Nonlinear Systems Identification. Springer, Berlin (2001)
9. Hastie, T.J., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer Science+Business Media, LLC, N.Y. (2009)
10. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. **15**(1), 116–132 (1985)
11. Jang, R.J.-S.: ANFIS: adaptive network based fuzzy inference systems. IEEE Trans. Syst. Man Cybern. **23**(3), 116–132 (1993)

12. Abiyev, R., Kaynak, O.: Fuzzy wavelet neural networks for identification and control of dynamic plants – a novel structure and a comparative study. IEEE Trans. Ind. Electron. **55** (2), 3133–3140 (2008)
13. Bodyanskiy, Y., Vynokurova, O.: Hybrid adaptive wavelet-neuro-fuzzy system for chaotic time series identification. Inf. Sci. **220**, 170–179 (2013)
14. Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. In: Advances Neural Information Processing Systems, pp. 524–532. Morgan Kaufman, San Mateo (1990)
15. Bodyanskiy, Y., Tyshchenko, O., Kopaliani, D.: A multidimensional cascade neuro-fuzzy system with neuron pool optimization in each cascade. Int. J. Inf. Technol. Comput. Sci. **6** (8), 11–17 (2014)
16. Bodyanskiy, Y., Tyshchenko, O.: Fast deep learning for cascade neuro-fuzzy system. In: 2nd International Workshop and Networking Event "Advances in Data Science", Holny Mejera, Poland, pp. 18–19 (2016)
17. Bodyanskiy, Y., Pliss, I., Peleshko, D., Vynokurova, O.: Adaptive deep learning of multivariate hybrid system of computational intelligence. In: VIIth International Conference on Decision-Making Theory, Uzhhorod, Ukraine, pp. 58–59 (2016). (In Ukrainian)
18. Wang, L.-X.: Adaptive Fuzzy Systems and Control: Design and Stability Analysis. Prentice Hall, Upper-Saddle River (1994)
19. Yamakawa, T., Uchino, E., Miki, T., Kusanagi, H.: A neo-fuzzy neuron and its applications to system identification and prediction of the system behavior. In: 2nd International Conference on Fuzzy Logic and Neural Networks, IIZUKA 1992, Iizuka, Japan, pp. 477–483 (1992)
20. Miki, T., Yamakawa, T.: Analog implementation of neo-fuzzy neuron and its on-board learning. In: Mastorakis, N.E. (ed.) Computational Intelligence and Application, pp. 144–149. WSES Press, Piraeus (1999)
21. Bodyanskiy, Y., Mulesa, P., Setlak, G., Pliss, I., Vynokurova, O.: Fast learning algorithm for deep evolving GMDH-SVM neural network in data stream mining tasks. In: 2016 IEEE First International Conference on Data Stream Mining and Processing (DSMP), pp. 257–262 (2016)
22. Bodyanskiy, Y., Setlak, G., Pliss, I.P., Vynokurova, O.: Hybrid neuro-neo-fuzzy system and its adaptive learning algorithm. In: Xth IEEE International Science and Technology Conference on Computer Science and Information Technologies, Lviv, Ukraine, pp. 111–114 (2015)
23. Lee, S.-H., Lee, J.-G., Moon, K.-I.: Smart home security system using multiple ANFIS. Int. J. Smart Home **7**(3), 121–132 (2013)
24. Bodyanskiy, Y., Peleshko, D., Setlak, G., Mulesa, P., Vynokurova, O.: Adaptive multivariate generalized additive neuro-fuzzy systems and its on-board fast learning. Neurocomputing **230**, 409–416 (2017)
25. Bodyanskiy, Y., Peleshko, D., Tatarinova, Yu., Vynokurova, O: Architecture of hybrid generalized additive neuro-fuzzy system in modelling technological process. In: XIIIth IEEE International Conference on The Experimental of Designing and Application of CAD Systems in Microelectronics, Lviv-Polyana, Ukraine, pp. 333–335 (2015)
26. Bodyanskiy, Y., Setlak, G., Peleshko, D., Vynokurova, O.: Hybrid generalized additive neuro-fuzzy system and its adaptive learning algorithms. In: The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Warsaw, Poland, pp. 328–333 (2015)
27. Landim, R.P., Rodrignes, B., Silva, S.R., Matos, W.: A neo-fuzzy-neuron with real-time training applied to flux observer for an induction motor. In: Vth Brasilian Symposium on Neural Networks, pp. 67–72. IEEE Computer Society, Los Alamitos (1998)

28. Silva, A.M., Caminhas, W., Lemos, A., Gomide, F.: A fast learning algorithm for evolving neo-fuzzy neuron. Appl. Soft Comput. J. **14**, 194–209 (2014)
29. Hastie, T.J., Tibshrani, R.J.: Generalized Additive Models. Chapman and Hall, London (1990)
30. Bossley, K.M., Brown, M., Harris, C.J.: Neuro-fuzzy model construction for modelling of non-linear processes. In: 3rd European Control Conference, Italy, vol. 3, pp. 2438–2443 (1995)

Fault Tolerant ASIC/ULA-Based Computing Systems Testing via FPGA Prototyping with Fault Injection

Oleg Brekhov(✉) and Alexander Klimenko

Department of Control Systems, Informatics and Electrical Engineering,
Moscow Aviation Institute (National Research University),
4 Volokolamskoe shosse, Moscow 125993, Russia
obrekhov@mail.ru, a.v.klimenko@mai.ru

Abstract. Testing is a significant part in the development of fault-tolerant systems on chip (SoCs) based on ASIC/ULA. This is due to complexity of the systems themselves and to the necessity to verify not only main functionality, but also efficiency of the implemented fault tolerance ensuring mechanisms. The FPGA-based fault injection method is widely used to speed up this process but its efficiency depends on implementation. This article discusses adequacy issues for the novel testing methodology for the fault-tolerant SoCs based on ASIC/ULA.

Keywords: Fault injection · FPGA-prototyping · ASIC to FPGA conversion · Fault tolerant system testing

1 Introduction

Testing is a crucial part of the modern ASIC/ULA based SoCs development life-cycle but growing devices densities and reduced elements sizes turn it into a significant challenge. Moreover, fault tolerance requirements which are increasingly imposed not only for special purpose systems (e.g. aerospace) that traditionally have high reliability requirements but also to commercial systems, make its own contribution to the great complexity of the testing. These requirements may be due to the need for a chip yield improvement [1] and the increased influence of external harsh environments owing to devices dimensions shrinking [2]. In our work we examine testing method only for SoCs with hardware implemented fault tolerance. However, complex computing systems may also use software approaches, such as [3].

The fault injection method is widely used for testing of fault-tolerant systems [4, 5]. However, the analysis of works in this field can define four basic approaches to its implementation [6, 7, 8]: (1) Hardware fault injection methods, which refer to the usage of special equipment that generates physical impacts on the prototype of the target system. Basic features of this group of methods include high cost, high speed, bad repeatability of the experiments and high probability of prototype corruption. (2) Software fault injection methods, which also can be applied only after production of a pilot batch. These methods implement fault-injection at the software level.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_6

(3) Simulation based fault injection methods. This group of methods involves creating a software model of the system and injecting faults in it. Basic features of this group of methods include wide opportunities for implementation of different fault and failure models, high visibility of the testing results, and low testing speed. (4) Emulation based fault injection methods. This approach usually implies creation of synthesizable system model with the help of hardware description languages and its further implementation as FPGA prototype to which fault injection is applied. It combines advantages of both hardware and simulation based methods enabling high-quality fault-tolerant SoCs testing in a short time. Testing time reduction plays a key role, since in some cases the cost of testing can reach up to 40% of the final product cost [9].

This article describes fault-tolerant SoCs testing method and its implementation via hardware-software complex (HSC), based on fault injection and FPGA prototyping. The particular implementation of fault injection method, used in HSC, is considered to review the functional compliance achievement between the system being developed and its FPGA prototype.

2 Proposed Method

In [10] we proposed hardware-software complex (HSC) for testing of fault tolerant computing systems on chip (see Fig. 1). The complex comprises of the workstation (or host computer) which runs HSC software and four PCIe expansion boards, each containing Xilinx Virtex-6 FPGA and used for conducting fault injection experiments.

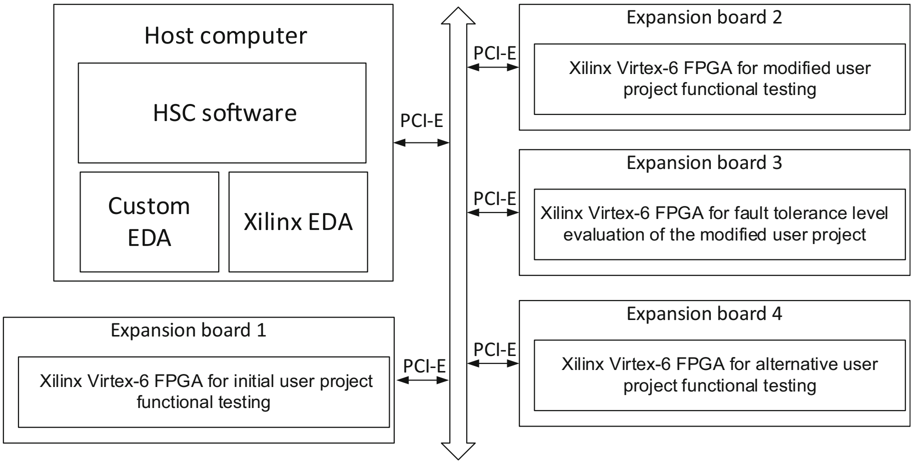

Host computer
HSC software
Custom EDA
Xilinx EDA
PCI-E
PCI-E
Expansion board 2
Xilinx Virtex-6 FPGA for modified user project functional testing
PCI-E
Expansion board 3
Xilinx Virtex-6 FPGA for fault tolerance level evaluation of the modified user project
Expansion board 1
Xilinx Virtex-6 FPGA for initial user project functional testing
PCI-E
PCI-E
Expansion board 4
Xilinx Virtex-6 FPGA for alternative user project functional testing

Fig. 1. The HSC block diagram.

The HSC implements multi-phase simulation methodology (also presented in [10]) that allows to perform both basic functionality testing and effectiveness evaluation of fault-tolerant ensuring means. It has an extended concept of fault injection method.

[11] as a basis and uses “fault appearance model” to simulate harsh operation environments during fault injection. This concept refers to transformation of chemical and physical changes of the chip, caused by impact of external or internal fault sources, into faults and failures in logic circuit of computing system. It was proposed to distinguish three main components of the fault appearance model: external influences model (EIM), threats occurrence model (TOM) and fault localization model (FLM). These three models form a stack (see Fig. 2) since the output of one model is used as input of another one. The EIM produces fault sources characteristics from the user-defined operation environment characteristics. The TOM then determines what physical and chemical effects such fault sources will cause in the chip. These changes are then finally converted by FLM to logic faults in computing system circuit. These faults are directly inserted during the fault injection.

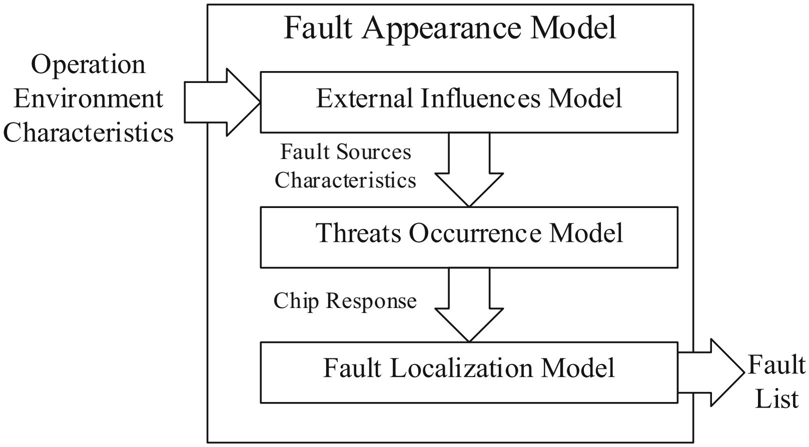


Fig. 2. Fault appearance model block-diagram.

The process of direct faults injection to the system elements is carried out by modification of their source code. Computing system is described at the gate level and is presented in the Verilog netlist format. Modification is made by replacing the description of certain elements of the system to their analogues with additional functionality for fault injection. This method refers to the class of code-modification techniques and to the group of saboteur-modules implementation methods, according to classification given in [4].

The distinctive feature of our fault injection method implementation is organization of saboteur modules control. We use a set of hardware modules, implemented together with the target computing system (user project) in one FPGA for that purpose. These modules form so-called “simulation support microkernel” (SSM) (see Fig. 3). The SSM implements the following functionality: fault injection, input actions providing, user project clock control, receiving of the project responses and interaction with HSC software for transferring simulation results and receiving control commands. Most of the work is done by the local control unit (LCU) which manages other SSM

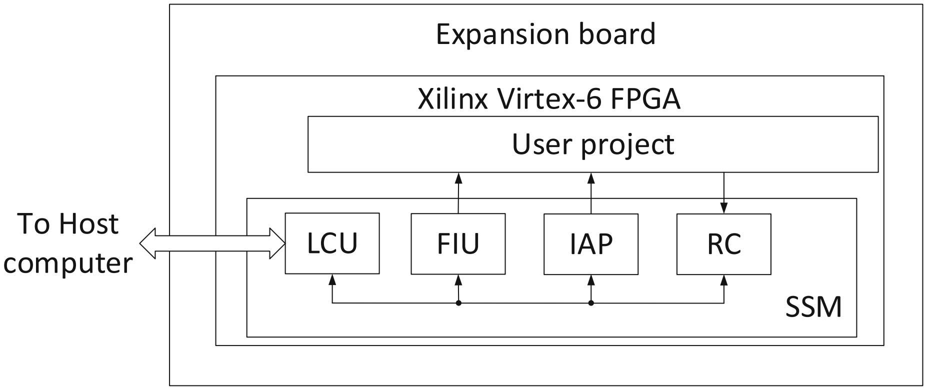


Fig. 3. "Simulation support microkernel" organization. SSM – simulation support microkernel; LCU – Local Control Unit; FIU – Fault Injection Unit; IAP – input actions provider; RC – response collector

units (such as Fault Injection Unit (FIU), Input Actions Generator (IAG) and Response collector, RC) and interacts with HSC software.

The testing methodology implemented in HSC uses the following input data:

- Computing system model (initial user project), presented in Verilog netlist format and in a target hardware basis (THB). It is produced by the special THB-oriented electronic design automation tool ("Custom EDA") which should be presented in HSC.
- List of circuit elements chosen by the user for fault injection.
- List of outputs of the internal circuit elements, which values should be under control during the testing (list of monitoring points).
- Fault appearance model (which is implemented as a dynamic link library, DLL) and its input data.
- THB primitives description library represented in Verilog/SystemVerilog and used to convert system project from THB to Xilinx FPGA basis.
- Xilinx EDA, allowing to form netlist for the system project and configure an FPGA in accordance with this netlist.
- Library of THB primitives with fault injection capability represented in Verilog/SystemVerilog.

All input data should be initialized by the HSC user before the testing begins. After that HSC consistently performs: (a) initial user project analysis; (b) user project migration to Xilinx FPGA basis and corresponding netlist file creation; (c) replacement of chosen circuit elements by their counterparts with fault injection capability in netlist file received in (b), and creation of corresponding netlist (so-called "modified user project"); (d) generation of two SSM netlist files (one with and one without fault injection capability), as well as their integration with netlists, obtained in (c) and (b) respectively.

Next, HSC generates the arrays of input actions and reference outputs. These arrays are used for functional testing of both the original and modified user projects. After that, HSC manages Xilinx CAD to configure three Xilinx FPGAs using netlists obtained in (d). The first FPGA contains initial user project and SSM without fault injection capability; the second and the third FPGAs both contain modified user project and SSM with fault injection capability.

After all previous actions HSC performs functional testing of the initial and modified user projects in the absence of faults. This phase of the methodology helps to ensure that projects are consistent. In case of testing results coincidence HSC manages fault appearance model to perform fault list generation. Current HSC version supports three basic fault models: memory element logical value inversion, forcing trigger to 0 and forcing trigger to 1. Fault appearance model firstly analyzes its input data and determines estimated operation conditions of the target SoC. Then it performs fault list generation by converting the possible chemical and physical changes in the chip to the faults of these three types.

The created fault list is then used for performance simulation of the modified project in the presence of faults (the third FPGA is used). If it turns out that the current fault tolerance level of modified project is not enough, the HSC user could test an alternative fault tolerance ensuring means in parallel, by implementing alternative user project and provide its functional testing using the fourth FPGA.

Simulation results allow to determine the effectiveness of fault tolerance means used in computing system.

The software part of the HSC is described in [12], and consists of five main components: simulation management component, component of system project processing, input actions generator, fault list generator and simulation results processor. In general, the HSC software provides the following functions: (1) functional analysis of the original system project (2) Detecting circuit elements, which are more likely to be affected. (3) Determining the consequences of fault occurrence. (4) Generating external influences for the system project. (5) Generating configuration data for FPGA based expansion boards. (6) Implementing performance simulation of the system project in the presence of faults. (7) Generating timing diagrams of the internal signals of the system project. (8) Performing the collection, analysis, storage and processing of the simulation data.

The hardware part of the HSC consists mainly of the SSM modules and expansion boards. It allows implementing both the original and modified system projects in FPGA and carrying out their functional testing in the absence or in the presence of faults.

3 Implementation Analysis

During the HSC development the following limitations have been taken: firstly, we assume that computing system being developed utilizes single ASIC/ULA chip. Another requirement is that the propagation delay of the critical path should be less than the clock period. In other words, if you look at the circuit of the system as a set of sequential and combinational logic, the signal on any combinational path have to spread from the beginning to the end during one clock period. These requirements simplify the prevention of possible functional discrepancies between computing system and its FPGA prototype.

The main factors of appearing of functional discrepancies between computing system and its FPGA prototype are the following: different clock frequencies in FPGA and target ASIC/ULA, different logical functions implementations, different voltage levels and different time of signal propagation through FPGA and ASIC/ULA elements (various delays of the elements).

In general, when using FPGA prototype for performance simulation of the target system in the absence of faults, the testing results adequacy could be reached by providing the same ratio of the signal propagation time between all corresponding combinational paths in target system and its FPGA prototype. In other words, signal propagation time between each two memory elements in logic circuit of the system represented in ASIC/ULA basis and propagation time between corresponding elements in such circuit represented in FPGA basis must be either equal or should relate with a certain proportionality factor. This factor must be the same for all logical paths. Ensuring the fulfilment of this requirement is sufficient to obtain reliable testing results if previous requirement for signal propagation is met (all signals should manage to propagate during one clock period).

The SSM was entrusted with clock control function for this purpose. It generates active clock edges for system project between sessions of input signals feeding to its inputs and reading values from its outputs. Thanks to this solution, it became possible to avoid functional inconsistencies during performance simulation in the absence of faults.

When we speak about functional conformity while performing simulation in the presence of faults, we should take into account the used fault models and models of fault sources.

The HSC was developed for the testing of fault-tolerant aerospace computing systems, which suffers mainly from cosmic radiation [13]. There are a lot of radiation fault models designed for different production technologies. However, regardless of fault type, it ultimately comes down to signal values changing in the device logic diagram. Among main types of logic faults we define faults in memory elements, such as logic value inversion, forcing trigger to zero or one and spurious pulses in combinational logic. In real systems, faults mainly occur during clock period between active edges, so the implementation of fault propagation models is of great importance.

Due to different signal propagation times for ASIC/ULA and FPGA and analog nature of faults in combinational logic it is inconvenient to implement such models in hardware. In this regard, it was decided to implement in HSC only three previously described fault models for the direct injection in system elements. But fault appearance model which is used for the fault list generation can comprise of software models, implemented in external DLL. These external models can be used for simulating faults in combinational logic and calculation of erroneous signal propagation (from the moment of fault occurrence until it would be triggered in registers, connected to this combinational path). The objectives of these models include faults occurrence localization, fault type determination, calculation of the erroneous signals propagation time until they would be triggered in memory elements or attenuated. A set of memory elements in which these erroneous signals would be triggered should also be determined as well as the vector of their values and the number of the corresponding modelling cycle. However, all calculations should be done for the THB.

The proposed approach gives an opportunity to simulate almost any kinds of faults by developing an appropriate models. Herewith, functional compliance between computing system and its FPGA prototype also provided, because the simulation of further propagation of triggered erroneous signal is similar to performance simulation in the absence of faults.

4 Conclusions

The article presents testing method for the fault tolerant computing systems on chip based on ASIC/ULA. Used software and hardware solutions allow to simulate almost any types of faults arising from different sources. The article describes the solution used for ensuring functional compliance between computing system and its FPGA prototype.

References

1. Kleihorst, R.P., Nieuwland, A.K.: IC cost reduction by applying embedded fault tolerance for soft errors. J. Electron. Test. Theory Appl. **20**, 533–542 (2004)
2. Nicolaidis, M.: Soft Errors in Modern Electronic Systems. Springer (2011)
3. Gawkowski, P., Rutkowski, T., Sosnowski, J.: Improving fault handling software techniques. In: Proceedings of On-line Testing Symposium, pp. 197–199. IEEE Computer Society (2010)
4. Ziade, H., Ayoubi, R., Velazco, R.: A survey on fault injection techniques. Int. Arab J. Inf. Technol. **1**(2), 171–186 (2004)
5. Shwetha, N., Sunil, T.D., Kurian, M.Z.: A survey on high speed fault injection module for fault detection processor on FPGA. Int. J. Adv. Trends Comput. Sci. Eng. **4**(1), 10–13 (2015)
6. Benso, A., Prinetto, P.: Fault injection techniques and tools for embedded systems reliability evaluation. Springer (2004)
7. Rohani, A., Kerkhoff, H.G.: A technique for accelerating injection of transient faults in complex SoCs. In: 2011 14th Euromicro Conference on Digital System Design (DSD) (2011)
8. Kooli, M., Di Natale, G.: A survey on simulation-based fault injection tools for complex systems. In: 2014 9th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (2014)
9. Wang, L.-T., Wu, C.-W., Wen, X.: VLSI Test Principles and Architectures Design for Testability. Elsevier (2006). ISBN 9780080474793
10. Brekhov, O., Klimenko, A.: Hardware-software simulation complex for FPGA-prototyping of fault-tolerant computing systems. Commun. Comput. Inf. Sci. **678**, 72–86 (2016)
11. Brekhov, O., Kordover, K., Klimenko, A., Ratnikov, M.: FPGA prototyping with advanced fault injection methodology for tolerant computing systems simulation. Distrib. Comput. Commun. Netw. **601**, 208–223 (2015)
12. Brekhov, O., Klimenko, A., Shdanov, A., Yakupov, A.: Implementation of experimental software prototype for control of fault tolerance of IC design. Nanoindustry **8**, 48–58 (2016)
13. Petersen, E.: Single Event Effects in Aerospace. IEEE Press (2011)

Critical Energy Infrastructure Safety Assurance Strategies Considering Emergent Interaction Risk

Eugene Brezhnev[1,2(✉)], Vyacheslav Kharchenko[1,2], Viacheslav Manulik[1], and Konstantin Leontiev[2]

[1] National Aerospace University KhAI, Kharkiv, Ukraine
e.brezhnev@csis.org.ua, V.Kharchenko@khai.edu,
viacheslav.manulik@gmail.com
[2] Research and Production Company Radiy, Kropyvnytskyi, Ukraine
ksleontiev@gmail.com

Abstract. Critical energy infrastructures (CEI) are the basis for the development of modern society. CEI consists of interacting systems that are integrated to implement the target function. There are interconnections between system safety states. Failures and accidents of CEI are characterized by the high severity of consequences. One of the causes of accidents (failures) in CEI is the imperfection of methods of risk analysis, in particular, not accounting for emergent risks (ER) associated with a negative interference between system safety states. There are two strategies for decreasing of ER in CEI: the use of diversity (off-line emergent risk management) and the reallocation of resources between systems (on-line emergent risk management). In this chapter, the problem of redistribution of CEI resources was formulated and two additional strategies to reduce ER were offered. The main difference between the offered strategies is based on individual preferences of CEI systems, in which redistribution takes into account (excludes) the capabilities of systems to decrease their ER and to ensure the required values of the safety indicators. In this article, the comparative analysis of the offered strategies was carried out.

Keywords: Critical energy infrastructures · Security · Redistribution of resources · Emergent risks · Interference

1 Introduction

According to [1], the critical energy infrastructure (CEI) is a set of interrelated service structures or objects, which form the basis of the functioning of modern society. In a broader sense, the term "infrastructure" refers to the people, organizations, processes, services, information flows and as well as technical installations and constructions that are included in the functioning of society, economy, and the state, individually or in a network. Analysis of the number of accidents [2, 3] associated with CEI, confirmed the relevance of the problem of safety. Safety analysis (risk) of CEI is carried out at all stages of its life cycle (LC). However, the initial phase (design of CEI) is characterized by a high uncertainty due to lack of awareness of the subject of analysis (the researcher)

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_7

about the risks, the modalities of CEI, the degree of mutual influence between systems, etc. This leads to uncertainty (inaccuracy) of the risk analysis and, as a consequence, problems (errors) in identifying, assessing and reducing the risks in CEI.

The interaction (informational, physical, geographical, etc.) between systems in CEI leads to new (emergent, hereinafter ER) risks that cannot be identified in the early stages of LC. If the local risk (LR) are measured and reduced at the design stage of the system, the uncertainty in the estimation of ER remains one of the main sources of danger to CEI and its safety. Thus, increasing the safety of CEI can be achieved by identifying and reducing ER.

It should be noted that the safety of CEI can be achieved by introducing diversity between systems [4]. However, this strategy is acceptable only at the design stage of CEI design (off-line risk management), when design solutions aimed at reducing overall vulnerabilities in CEI are proposed and implemented. However, when ER arise during combining systems in CEI, a strategy of diversity may not provide the required level of safety (LS). It is, therefore, advisable to consider some resource T (tangible assets, percentage of the total load) for each system which can be used to reduce ER when operating CEI (on-line risk management). This assumption is based on the analysis of the accident [2], in which the systems of CEI were able to increase its contribution to the regulation of overall load, distributing it among them. The more total load (generation of electricity as a service) the system has, the more its own resources it should allocate for CEI as a whole (in fact, to pass her).

Since each of the CEI systems has some resources (ability to load-sharing), then one of the possible strategies to reduce ER is the redistribution aimed at the use of the excess resources of one system to reduce ER of another system, due to the interaction.

Currently, there are many approaches to the reallocation of resources in the systems. In [5] an analysis of methods of resource allocation and the possibility of their use in Grid systems are presented. In [6] a general mathematical model of dynamic scheduling in a distributed computing system is described. It is shown that the task assignment to the computing resource is reduced to the problem of finding a maximum matching in a bipartite graph. In [7] we can see a mathematical model and method of dynamic allocation of scarce resources to the system computer applications in the aggregate, which takes into account the needs and priorities of each application with varying user load on them. In [8] the development of the Internet as a dynamic stochastic network, which is characterized by the spatial distributions of information resources and information flows, is shown. In [9] we offer the standpoint of control of network resources dynamic mathematical models of telecommunication systems in terms of the adoption of various hypotheses about the nature of the parameters of the system and completeness of initial information about them. Attempts for creative solutions to tasks involving alternative planning tools are made. So, in [10] game theoretical models of spectrum sharing in wireless cognitive radio networks (spectrum is an available resource) are considered. In [11] it was proposed to use the reserve funds to cover unexpected expenses and to establish the correlation between the potential risks and costs necessary to overcome the effects of these risks. In the general case, the reserve is used to finance additional work, compensation for unforeseen changes in material and labor costs, overhead costs and other expenses arising in the course of the project.

Thus, the main disadvantages of the offered methods are: ignore of ER during deployment; ignore of differences in the target function of each system, which can be allocated to the individual preferences of each of the system resources to other systems; ignore of interference between systems; systems are not treated as a single unit, which leads to a lack of formalization of a common target function for the system as a whole.

The purpose of the article is the formation of safety strategies of CEI through the reallocation of resources between its systems to reduce ER.

2 Safety Strategies: Redeployment of Resources in CEI

All systems in CEI are open, i.e. systems which in the course of their LC exchange resources (power loadable reserve, etc.) and information with other systems. The degree of openness (DO) is characterized by many parameters: number of ties with other systems type, period of interaction. DO of any CEI system changes as all the parameters of the interaction change. Open systems are subject to the negative influence of other systems in CEI. The more the system is, the greater its vulnerability to negative influence.

The results of risk analysis of CEI depend on the adopted (under studies) system DO. To determine DO, especially at the stage of project analysis, is quite difficult. The underestimation of interaction makes the system in some sense conservative “closed”. This leads to the fact that risk analysis becomes inaccurate, deterministic, which reduces the reliability of the results and effectiveness of countermeasures aimed at reducing ER. Please do not include section counters in the numbering.

CEI can be represented as a set of interacting systems (see Fig. 1), linked by bonds of different nature (electricity flow, information, logical, etc.). In the general case, CEI (S_0) can be represented as a set of interacting systems. Risks in open and closed (without interaction) systems are different.

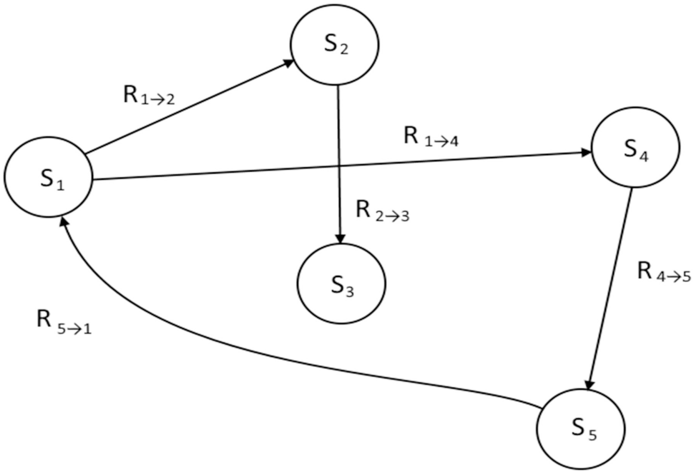


Fig. 1. General view of CEI S_0 and the interaction between it systems

In the general case, the total amount of LR for a closed system S_1 (without taking into account the interference from other systems in CEI) at time t can be represented by the additive convolution of the form:

$$R_{S_1}^{Closed}(t) = \sum_{i=1}^{N} \alpha_i R_i(t), \tag{1}$$

where $R_{S_1}^{Closed}(t)$ - cumulative LR for the closed-loop system S_1 at time t;

$R_i(t)$ - specific LR for S_1 at time t;

α_i - coefficient describing the priority of LR for the system.

We can assume that for system S_1, some CEI component part of S_0, in the presence of interference between S_1 and S_2, the magnitude of the resulting risk to system S_1 is related to the S_2 system will be greater than the magnitude of the risk for S_1 closed the effect of the system S_2. In other words, the cumulative LR for the closed-loop system S_1 is less than its total risk as part of a system S_0, containing systems S_1 and S_2. The difference between the total risk of a closed and open system can be represented as:

$$R_{S_1}^{emerg}(t) = R_{S_1}^{S_0}(t) - R_{S_1}^{Closed}(t), \tag{2}$$

where $R_{S_1}^{emerg}(t)$ - the value of ER that occurs in the system S_1 as a result of interaction with S_2 in the system S_0;

$R_{S_1}^{S_0}(t)$ - the risk of system S_1 to the system CEI (S_0).

We can assume that for system S_1, some CEI component part of S_0, in the presence of interference between S_1 and S_2, the magnitude of the resulting risk to system S_1 is related to the S_2 system will be greater than the magnitude of the risk for S_1 closed the effect of the system S_2. In other words, the cumulative LR for the closed-loop system S_1 is less than its total risk as part of a system S_0, containing systems S_1 and S_2. The difference between the total risk of a closed and open system can be represented as:

$$R_{S_1}^{emerg}(t) = R_{S_1}^{S_0}(t) - R_{S_1}^{Closed}(t), \tag{3}$$

where $R_{S_1}^{emerg}(t)$ - the value of ER that occurs in the system S_1 as a result of interaction with S_2 in the system S_0;

$R_{S_1}^{S_0}(t)$ - the risk of system S_1 to the system CEI (S_0).

Thus, at any point in time, the system Sj is characterized by the value of LR and ER due to the negative influence of h-type (informational, physical, etc.) from another system Si. Assume that each system Si has a resource (reactive power) which can be used to reduce LR.

In the general case, the task of redistribution of resources within the CEI to reduce ER caused by interference between the systems can be formed. A shared resource of CEI is an additive amount of system resources, which is a limitation in redistribution.

Condition. The system may not transfer the resources if the current safety index (SI) and the value of ER do not match the required values.

Assumption. These resources are only used to reduce ER.

In general, the task of reallocating resources in CEI can be formulated as:

- there is some system $\mathrm{S_i}$ in CEI, with resources $\mathrm{M_{Si}}$. Safety Index and ER do not meet the required values, i.e. $\mathrm{CEI} = \left\{\{\mathrm{S_i}\}_\mathrm{I}, \{\mathrm{M_{S_i}}\}, \{\mathrm{SI_{S_i}}(\mathrm{t}) \notin \Omega_{SI^{accept}_{Si}(t)}\},\ \{R^{emerg}_{Si}\right.$ $\left.(\mathrm{t}) \notin \Upsilon_{R^{accept}_{Si}(t)}\}\right\}$;
- it is necessary to provide an acceptable level of LS and ER system by redistributing resources within CEI, i.e. $\mathrm{CEI} = \left\{\{\mathrm{S_i}\}_\mathrm{I}, \{M^*_{Si}\}, \{\mathrm{SI_{S_i}}(\mathrm{t}) \in \Omega_{SI^{accept}_{Si}(t)}\}, \{R^{emerg}_{Si}(\mathrm{t}) \in \right.$ $\left.\Upsilon_{R^{accept}_{Si}(t)}\}\right\}$ subject to the restrictions that $M_{CEI} = \sum_{i}^{I} M_{Si}$.

Within CEI there are various strategies of redistribution of resources to reduce ER.

3 Strategy of Redistribution with the Mandatory Allocation of Sufficient Resources

The system (the subject of influence, donor system) transfers to another system (object of influence) the amount of resource as necessary to reduce ER that it creates. In this case, the donor system must allocate a number of resources sufficient to maintain SI of another system (As Safe As Reasonably Practical, ASARP). This approach may not be rational for the donor system, because there are risks of situations in which it will not be able to reduce its ER, due to the influence of other systems. Object the influence of this approach is rational because required SI is provided.

Condition. The donor system Si can allocate resources to reduce ER just in case, if the current SI and the level of ER are acceptable, i.e. $\mathrm{SI_{S_i}(t)} \notin \Omega_{\mathrm{SI^{accept}_{si}(t)}}, \mathrm{R^{emerg}_{S_i}(t)} \in \Upsilon_{\mathrm{R^{accept}_{Si}(t)}}$.

The critical system condition Crt ($\mathrm{S_i}$) is considered as SI.

Assumption. The current resource system provides a reduction of its LR, i.e. the system must have resource is not less than that required to reduce the LR.

We introduce an indicator which characterizes the vulnerability of the resource (resource vulnerability index, RVI) systems in CEI:

$$RVI_{Si} = \frac{N_{Si \to Sj}}{N_{Sj \to Si}}, \tag{4}$$

where $\mathrm{N_{Si \to Sj}}$ - the number of outgoing ties (the system $\mathrm{S_i}$ is the subject of influence);

$\mathrm{N_{Sj \to Si}}$ - the number of incoming ties (the $\mathrm{S_i}$ system is the object of influence). The higher the value of RVI is the more risks of resource insecurity system in CEI.

We introduce an additional indicator – the ratio of the current LS Crt (Si) to the value of RVI – overall vulnerability index (OVI) of the system. The smaller OVI is the more vulnerable the system (low safety and high risks of resource insecurity).

The reallocation of resources within the framework of the first strategy includes the following steps:

1. Evaluation of ER $R_{S_i}^{emerg}(t)$, current SI Crt (Si) of systems and resource vulnerability index Li, OVI.
2. Ranking of CEI systems to a minimum of OVI with the purpose to select the system for the initialization of redistribution. Note that the distribution of resources between systems in CEI begins with systems with a minimum value of OVI. This means that the system has a low safety and high risks of insecurity. Thus, from the ranked series we take the vulnerable system, from which the improved safety is begun.

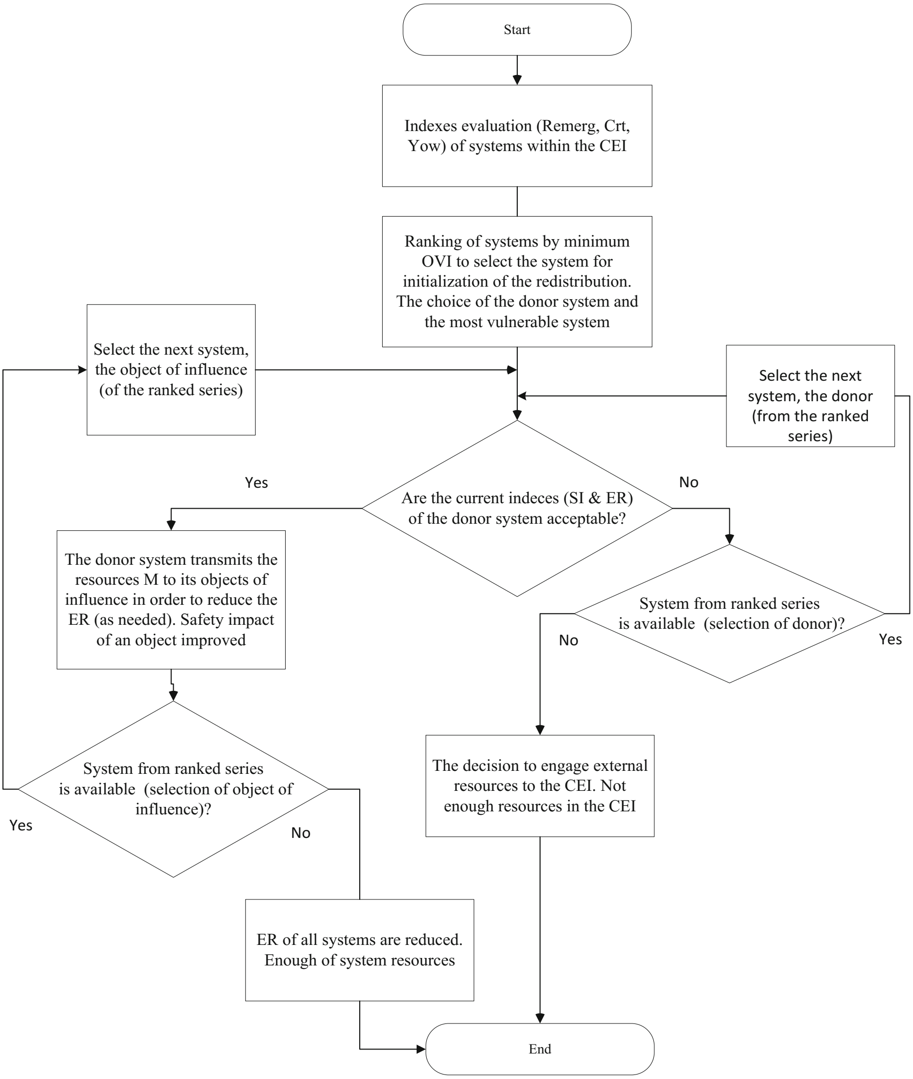

Start
Indexes evaluation (Remerg, Crt, Yow) of systems within the CEI
Ranking of systems by minimum OVI to select the system for initialization of the redistribution. The choice of the donor system and the most vulnerable system
Select the next system, the object of influence (of the ranked series)
Select the next system, the donor (from the ranked series)
Yes
No
Are the current indeces (SI & ER) of the donor system acceptable?
The donor system transmits the resources M to its objects of influence in order to reduce the ER (as needed). Safety impact of an object improved
System from ranked series is available (selection of donor)?
No
Yes
System from ranked series is available (selection of object of influence)?
Yes
No
The decision to engage external resources to the CEI. Not enough resources in the CEI
ER of all systems are reduced. Enough of system resources
End

Fig. 2. The first strategy algorithm

3. Determination of possibilities of the systems-subjects (associated with the affected system) for the transfer of resources to reduce ER system with a minimum OVI. If you have multiple subjects of influence, the transfer of resources for the object of influence begins from the "strong" system (the highest of OVI). Transfer of resources from the donor system of resources (with a minimum of OVI) is performed only if the current ER and the indicator of the safety of the donor are acceptable. The choice of the donor of resources system.
4. The resource transfer by the donor system for its objects of influence with the aim of reducing ER (subject to the requirements of LS and ER). Because the donor system transfers as much as you need, ER system of an object of influence is reduced after the transfer of resources.
5. Further, we considered the following system in ranked-set specified in step 2. Step 3 is repeated.
6. In the case when the donor system is not satisfied the conditions of admissibility of LS and ER, the transmission resources are not performed. Next, we study the following donor system (including ties) for the object of influence.
7. Transfer of resources is over when: it is impossible to transfer resources of one of the systems in CEI or when the conditions of admissibility of ER and SI for all systems in CEI.

Thus, each donor system must provide the resources to its object of influence, sufficient to reduce its ER. The algorithm of implementation the first strategy is shown in Fig. 2.

4 Strategy of Redistribution with the Possible (but Insufficient) Resource Allocation

The donor system transfers to the object of influence as many resources as possible, with a view to ensuring the required level of SI (with the existing ER relating to the negative influence of other systems), i.e., it allocates a certain surplus of resources, leaving itself just exactly what it needed.

This strategy is acceptable for the donor system because the remaining resource is sufficient to reduce its ER. For the system-object of influence, this approach may not be rational, as the allocated resources may be insufficient to reduce the ER created by the donor system.

Condition. The donor system Si may allocate resources only if its current SI and ER are acceptable, i.e. $\mathrm{SI}_{\mathrm{S_i}}(\mathrm{t}) \notin \Omega_{\mathrm{SI}_{\mathrm{si}}^{\mathrm{accept}}(t)}$, $\mathrm{R}_{\mathrm{Si}}^{\mathrm{emerg}}(\mathrm{t}) \in \Upsilon_{\mathrm{R}_{\mathrm{Si}}^{\mathrm{accept}}(\mathrm{t})}$.

Assumption. The current resource of a system provides the reduction of LR of the donor system.

It should be noted that in deployment (for both strategies), there are risks in which the donor system will give more resources than you get from the other systems. For example, the system S_4 (see Fig. 1) should give part of its resources to the system S_1 and S_5. System S_1 receives resources from the systems S_4 and S_5, transferring the portion of the resources of the system S_2. The larger the index of RVI system is,

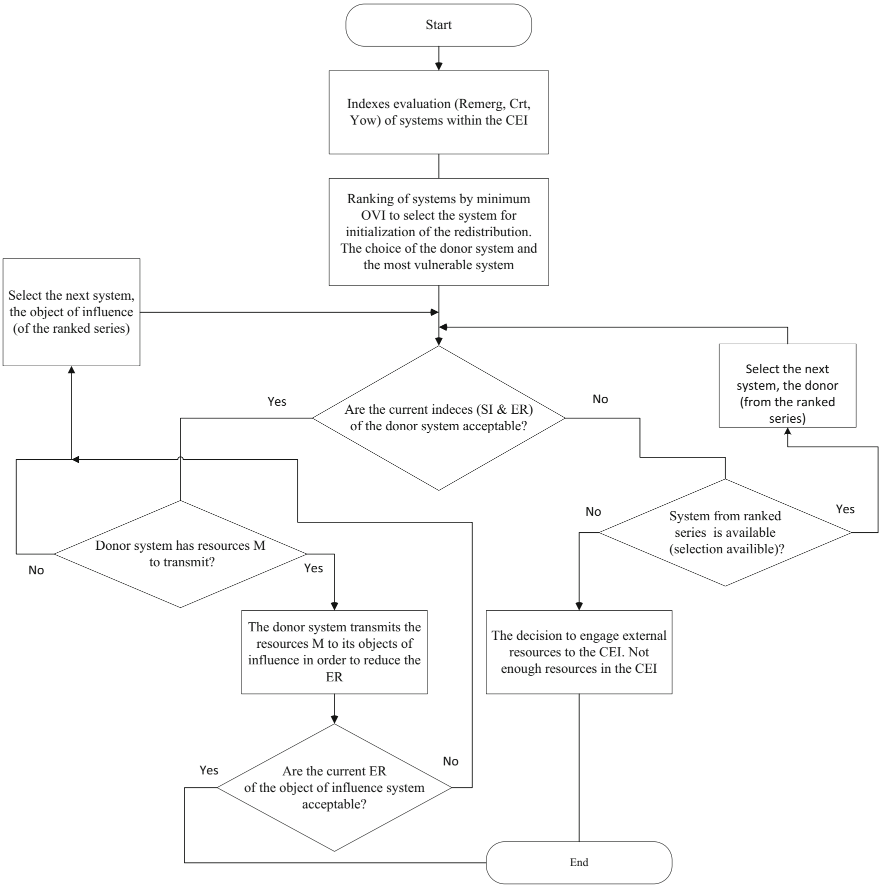

Start
Indexes evaluation (Remerg, Crt, Yow) of systems within the CEI
Ranking of systems by minimum OVI to select the system for initialization of the redistribution. The choice of the donor system and the most vulnerable system
Select the next system, the object of influence (of the ranked series)
Select the next system, the donor (from the ranked series)
Are the current indeces (SI & ER) of the donor system acceptable?
Yes
No
Donor system has resources M to transmit?
No
Yes
System from ranked series is available (selection availible)?
No
Yes
The donor system transmits the resources M to its objects of influence in order to reduce the ER
The decision to engage external resources to the CEI. Not enough resources in the CEI
Are the current ER of the object of influence system acceptable?
Yes
No
End

Fig. 3. The second strategy algorithm

the greater the risk of its insecurity resource to reduce LR and ER. The algorithm of implementation of the second strategy is shown in Fig. 3.

There are two special cases in the framework of the second strategy:

- resources M_{Si}^{**}, transferred by the donor system are not enough for full compensation of the ER, i.e. only some of the same resources, the size of which does not compensate for ER, are transferred. This means that you must use the following donor system (if any);
- resources transferred to another system fully ensure the reduction of ER. This case reduces to the first approach, therefore only the first case is further considered.

The second strategy is more complicated as to reduce ER object system of influence, can be used resources of all systems of the donor (including ties), which could lead to a situation where the donor systems can reduce their own ER (because they are subject to influence from other systems). For the first strategy, the number of possible donors will be less because it is defined more strict requirements on the transfer of resources to another system.

Let's consider the illustrative example of the implementation of two strategies. The source data for the implementation of the strategies are shown in Table 1.

Table 1. Initial data for the strategies implementation

System	Local risk	Emergent risk	System resource	Safety index, SI	RVI	SI/RVI, OVI
S_1	$R_{S_1}^{Closed}(t)$	$R_{S_1}^{emerg}(t)$	M_{s1}	Crt (S1)	RVI1	L1
S_2	$R_{S_2}^{Closed}(t)$	$R_{S_2}^{emerg}(t)$	M_{s2}	Crt (S2)	RVI2	L2
S_3	$R_{S_3}^{Closed}(t)$	$R_{S_3}^{emerg}(t)$	M_{s3}	Crt (S3)	RVI3	L3
S_4	$R_{S_4}^{Closed}(t)$	$R_{S_4}^{emerg}(t)$	M_{s4}	Crt (S4)	RVI4	L4
S_5	$R_{S_5}^{Closed}(t)$	$R_{S_5}^{emerg}(t)$	M_{s5}	Crt (S5)	RVI5	L5

In the present example (see Fig. 1) we obtained: final redistribution of risks and resources (first strategy, see Table 2); final redistribution of risks and resources (second strategy, see Table 3). A specific feature of CEI is to conduct risk management for all CEI in general.

Table 2. The final reallocation of risks and resources for the first strategy

	Resource state, T_0	Resource state due to its redistribution, T_1	The ability to fend off the ER	The ability to fend off the LR
S_1	M_{S1}	$M_{S1} + M_{S5}^{*} - M_{S1}^{*} - M_{S1}^{**}$	+	–
S_2	M_{S2}	$M_{S2} + M_{S1}^{*} - M_{S2}^{*}$	+	–
S_3	M_{S3}	$M_{S3} + M_{S2}^{*}$	+	+
S_4	M_{S4}	$M_{S4} - M_{S4}^{*} + M_{S1}^{**}$	+	–
S_5	M_{S5}	$M_{S5} + M_{S4}^{*} - M_{S5}^{*}$	+	–

Table 3. The final reallocation of risks and resources for the second strategy

	Resource state, T_0	Resource state due to its redistribution, T_1	The ability to fend off the ER	The ability to fend off the LR
S_1	M_{S1}	$M_{S1} + M_{S5}^{**} - M_{S1}^{**} - M_{S1}^{***}$	–	–
S_2	M_{S2}	$M_{S2} + M_{S1}^{*} - M_{S2}^{*}$	–	–
S_3	M_{S3}	$M_{S3} + M_{S2}^{*}$	–	+
S_4	M_{S4}	$M_{S4} - M_{S4}^{*} + M_{S1}^{**}$	–	–
S_5	M_{S5}	$M_{S5} + M_{S4}^{*} - M_{S5}^{*}$	–	+

The decrease in ER for a particular system is due to the allocation of resources across CEI. For example, the first strategy is implemented in two stages: (1) revealed a system with low-level LS and a high level of risk of resource failure; (2) this system provides resources to other systems (subject to the restrictions on relations), provided that its ER can be parried with the available resources of these systems.

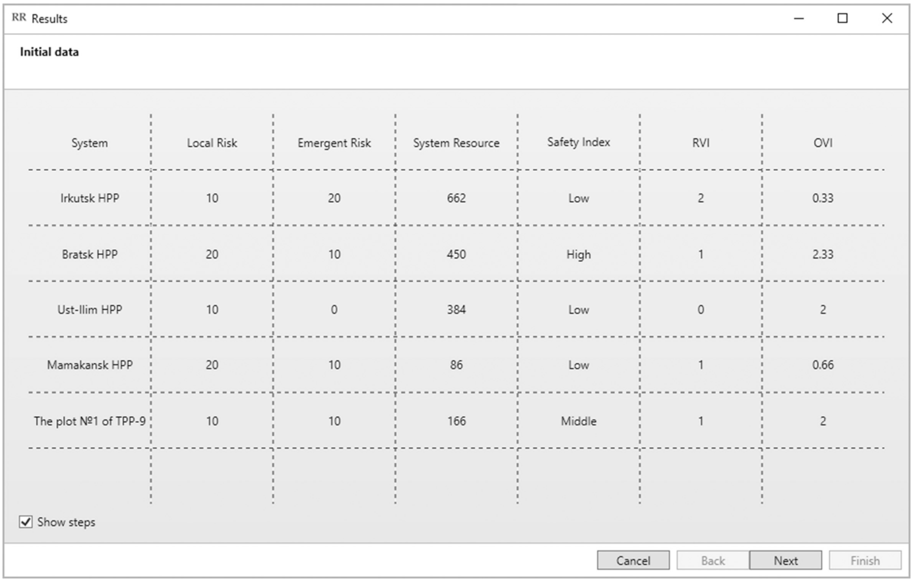
Comments. When using the first strategy the common ER of all CEI are balanced, because the systems transmit their resources to reduce them. In this case, if the system originally allocated the excess resources to reduce LR (with some margin), then LR are balanced in CEI.

It is obviously, that the second strategy does not only allow you to reduce ER but may not reduce LR for individual systems (provided that the transferred resources are not sufficient to mitigate ER). In this regard, CEI cannot provide the desired level of LS. For CEI, it is advisable to allocate additional resources.

A software tool has been developed to support this method. This program gives an ability to define system structure, describe elements state and predict a better strategy of resource redistribution as a result of calculations. The software tool was developed on top of Microsoft.NET platform using WPF framework to achieve a fast result and a high user experience. It gives an ability to easily create extensions and improvements to an application as an applied realization of the method.

4.1 Illustrative Example for Strategies Implementation

As a practical case of strategy application, the safety modeling of Siberia energy grid was performed (see Fig. 4). All generation systems (S_1–S_5) pull electricity into grid according to their capacities (resources). The systems' resources are their power (MW) capacity. ER for each station is a threat of event of taking additional power load in case when other systems will be lost due to accidents. System might be not able to take this additional load and the whole grid will be lost (black out). OVI is calculated after defuzzification of SI.



Initial data

System	Local Risk	Emergent Risk	System Resource	Safety Index	RVI	OVI
Irkutsk HPP	10	20	662	Low	2	0.33
Bratsk HPP	20	10	450	High	1	2.33
Ust-Ilim HPP	10	0	384	Low	0	2
Mamakansk HPP	20	10	86	Low	1	0.66
The plot №1 of TPP-9	10	10	166	Middle	1	2

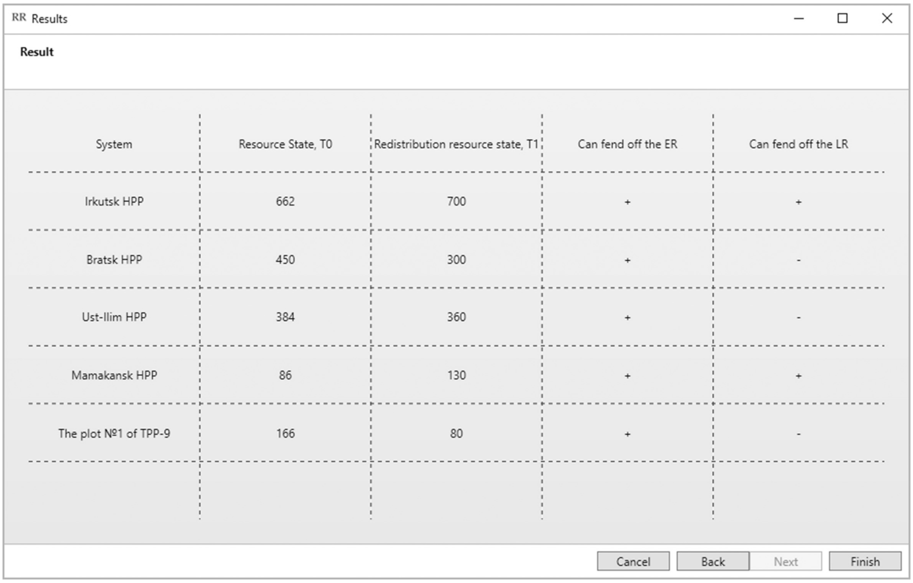
☑ Show steps

Fig. 4. The initial data for the first strategy

All the systems have a reserve capacity that can be allocated to support the balance of power grid (the balance of production and consumption). ER associated with the possible exclusion of one of the stations from the process of power generation.

The remaining systems will have to take the burden if it happens. If the station reserve has been decreased, it means that the system took over a part of someone's load. Physically - it started additional energy generation (for example, a generator for a power plant). If the reserve was increased, it should turn off these facilities, because the other system has taken over the part of the total load.

Analysis of final data (see Fig. 5) shows that after resource distribution only two generation stations (Irkusk and Mamakansk HPPs) improve their capacity to cope with ER, but decreased their capacity to cope with LR. If such strategy was considered by Syberia operator during decision-making likely Sayano–Shushenskaya HPP accident would have been avoided as this be based on the station abilities to decrease ER.



System	Resource State, T0	Redistribution resource state, T1	Can fend off the ER	Can fend off the LR
Irkutsk HPP	662	700	+	+
Bratsk HPP	450	300	+	-
Ust-Ilim HPP	384	360	+	-
Mamakansk HPP	86	130	+	+
The plot №1 of TPP-9	166	80	+	-

Fig. 5. The final data for the first strategy

5 Conclusions

Thus, the strategies implemented by reallocating the resources between systems in CEI, allow you to provide the required level of SI and ER with constraints on resources. Each of the offered strategies takes into account individual preferences in CEI systems: to provide as many resources as needed or as possible, taking into account the assessment of individual risks. The first strategy is preferable in incensement of risk of accidents (failures) associated with the (it is necessary to increase SI), the second involves the functioning of CEI in the moderate risk. Thus, the choice of strategy is due to the current SI systems, dynamics of interaction, resource safety of CEI. With the increasing of risks of accidents, it is advisable to use the first strategy. Monitoring of LR and ER will allow you to flexibility choose a particular strategy.

It is advisable to develop to support decision-making and implementation approaches. The implementation of the proposed approaches to the reallocation of resources is supported by the tool. This tool allows simulating the redistribution taking into account the characteristics of the systems included in CEI (strategy, initial amount of resources, communications, LR and ER).

References

1. Theocharidou, M., Giannopoulos, G.: Risk assessment methodologies for critical infrastructure protection. Part II: A new approach. Joint Research Centre. http://publications.jrc.ec.europa.eu/repository/bitstream/JRC96623/lbna27332enn.pdf. Accessed Dec 2016
2. Kut'in, A.G.: Technical Investigation Act of causes of the accident at the Sayano-Shushenskaya HPP. http://tayga.info/documents/3270/. Accessed Nov 2009
3. Tianyou, L., Juanjuan, L.: Analysis of icing accident in south china power grids in 2008 and its countermeasures. In: 20th International Conference and Exhibition on Electricity Distribution CIRED 2009, Prague, Czech Republic, pp. 1–4 (2009)
4. Brezhnev, E.V.: The method of choice of rational diversity strategy IMS of NOO in the face of uncertainty. Control Navig. Commun. Syst. **2**(23), 156–159 (2012)
5. Volk, M.A., Fylymonchuk, T.V., Gridel, R.N.: Methods of resource allocation for grid-systems. Sci. Works Kharkiv Air Force Univ. **1**(19), 100–104 (2009)
6. Symonenko, V.P.: The mathematical formulation of the problem of work dynamic allocation in GRID systems and evaluation of solution quality. In: Proceedings of the National Technical University of Ukraine "Kyiv Polytechnic Institute". Series: Informatics, Management and Computing, vol. 53, pp. 37–41 (2011)
7. Matveev, G.A., Trushkova, E.A.: Model of dynamic resource allocation. J. Buryat State Univ. **9**, 274–279 (2011)
8. Popkov, Yu.S.: Macrosystems and GRID-technologies: simulation of dynamic stochastic network. Issues of management. **88**, 10–20 (2003)
9. Popovskij, V.V., Lemeshko, A.V., Evseeva, O.Yu.: Telecommunication systems resources dynamic control: the state-space based mathematical models. Sci. Notes UNDIZ. **1**(9), 3–26 (2009)
10. Oshmarin, D.V.: Distribution of channel resources in cognitive radio networks based on game theory. Bus. Inform. **4**, 38–45 (2010)
11. Gorfinkel, V. Ya. (ed): Business Economics: Textbook. JuNITI, Moscow (2007)

Modelling an Optimal Capital Structure of the Telecommunication Company

Alexandr Y. Bystryakov, Tatiana K. Blokhina, Elena V. Savenkova, Oksana A. Karpenko(✉), and Elena V. Ponomarenko

Economics Department of Peoples' Friendship University of Russia,
ul. Miklukho-Maklaya 6, Moscow 117198, Russia
{bystryakov_aya, blokhina_tk, savenkova_ev, karpenko_oa, ponomarenko_ev}@pfur.ru, karpenox@list.ru

Abstract. The article is devoted to the development and approbation of a methodological approach to the modelling of an optimal structure of a joint stock company. The optimal capital structure is defined by the authors as a combination of debt and equity, which maximizes the overall value of the company. The article contains the main conclusions received from different economic researches on the optimization of a capital structure. The authors invented a model of the optimal capital structure that may be used by the joint stock companies in emerging markets with greater risk due to political instability, domestic infrastructure problems, currency volatility and limited equity opportunities. This model is supplemented with corrective indicators of financial risks.

Keywords: Capital structure · Debt and equity · The cost of capital · Financial risks · Risk management

1 Introduction

Statement of the problem. The capital of the joint stock company determines a potential production (trade, services) during activities, and finances its development. Decisions of the management always influence on the capital structure of the company. The capital structure also has a direct impact on the market value of the company which is determined by a valuation of its discounted future cash flows which depend beside with the choice of the discount rate based on profitability level and on the expected risk. Therefore, the main objective of the financial manager is the maximization of return on the asset in the adopted level of financial risk in order to raise the value of business. That is why the optimization of a capital structure is a fundamental process in a business management.

The theory of finance is not able to determine a universal formula which would allow constructing an optimal capital structure for all companies. Therefore the authors of this article have set the objective to systematize the main conclusions received from different economic researches on the formation and optimization of a capital structure and find the model of an optimal capital structure suitable for emerging countries' joint stock companies.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_8

The purpose of the study is to develop an integrated approach to the assessment of the optimal capital structure and of the debt-to-equity ratio, which allows reducing the financial risks of the company and maximizing the value of the corporation. This integrated approach may be used by the joint stock companies in emerging markets, which have greater risk due to political instability, domestic infrastructure problems, currency volatility and limited equity opportunities.

The object of the research is a capital structure of a Russian telecommunication joint stock company, mobile telesystems.

Main results of the study. The comprehensive review of the models of an optimal capital structure, including some very interesting empirical approaches, was done. Within the current situation in emerging markets, every theory and model for determining the optimal capital structure may be used only with restrictions and corrections for the effect of financial risks. The main criteria for determining the optimal capital structure while using different simulations of the capital structure are: acceptable levels of return and risk in the company's activity; minimization of the weighted average cost of capital of the company; maximization of the market value of the company.

The authors proposed a model for the determination of the optimal capital structure. They used the original model designed by A. Damodaran [2] and added several other author's assumptions, including the influence of modern risk factors (credit, interest rate, and equity risk).

2 Theoretical Background

The concept of "optimal capital structure" reflects the certain ideal ratio of a debt and an equity increasing the value of the company. It is very difficult to estimate the value of this ideal ratio of equity and debt capital. Therefore, managers of the company, while using various high-quality and quantitative methods, can only determine a certain target objective of a capital structure for the company which, in their opinion, will promote its successful development.

The optimal capital structure is defined by the authors as a combination of debt and equity, which maximizes the overall value of the company.

Currently, there are two main theories that compete with each other, static and hierarchical theories of capital structure [4]. The static theory of capital structure sets a target ratio of debt to asset value, and the company adjusts dividends by gradually moving in this direction.

The hierarchical theory of capital structure is presented in the form of financial hierarchy, where the company prefers domestic financing to an external one and debt to equity (equity issuance) [1]. In the hierarchical theory, the company has no distinct target ratio of debt to asset value. The capital structure of an enterprise is usually considered as the ratio between different sources of capital (equity and debt capital) used to finance its activities). Sometimes short-term borrowings are excluded from the capital, thus the capital structure is defined as a total of sources used to finance the investment activity of the enterprise in the long term [10]. However, in cases where short-term borrowings are made on a regular basis (which is mostly the case), in our opinion, they should be included in the capital composition in the analysis of the financing structure.

Nowadays, there is no consensus among the economists concerning the model of an optimal capital structure. Most of them perform an empirical analysis (using regression equations), introducing proxy variables which correlate with the major factor of the model, for instance, with the leverage level. Some financial economists, particularly Murray Frank, Vidhan Goyal, Mark Leary, Michael Roberts, Christopher Hennessy, and Toni Whited [5, 6] have offered very interesting empirical approaches to solving the mystery of capital structure (rebalancing the capital structure, return on the actual value of the capital structure to the average level (tests of mean reversion) in a relatively short period, dynamic capital structure models). Nevertheless, such issues as a company's criteria for the selection of debt, equity (common shares) or hybrid securities (preference shares) remain debatable when addressed [3]. The main direction of researches is a definition of the capital structure of an enterprise which is usually considered as the ratio between different sources of financing its activities. Sometimes short-term borrowings are excluded from the capital, thus the capital structure is defined as a total of sources used to finance the investment activity of the enterprise in the long term [10]. However, in cases where short-term borrowings are made on a regular basis (which is mostly the case), in our opinion, they should be included in the capital composition in the analysis of the financing structure.

If we approach the issue of determining the optimal capital structure from the perspective of the relative cost of financing sources, it is necessary to take into account that debt is cheaper than shares. Therefore, the price of debt capital is lower than the price of equity. Consequently, replacement of shares by the cheaper debt capital reduces the weighted average capital cost of capital, which results in an increase of business efficiency and, therefore, maximizes the value of the enterprise. Moreover, a number of financial management theories are based on the conclusion that the optimal capital structure involves the use of debt capital to the maximum possible extent. For example, several theories take into account agency costs. The Jensen's model defines the optimal capital structure as the structure when all the benefits of debt cover the cost of debt [8]. M. Harris and A. Raviv affirmed that the optimal capital structure should support the balance between debt, benefits and information costs to serve them [7].

However, in practice, one should take into account that the replacement of shares by the cheaper debt capital reduces the value of the company, which is determined by the market value of the company's equity. In addition, the growth of the debt increases the risk of bankruptcy, which could significantly affect the price that potential investors will agree to pay for the common shares of this company.

The use of debt capital also involves such important non-financial costs as the result of restrictions on the freedom of actions of managers in loan agreements. These may be obligations of creating additional reserves for debt repayment or limiting conditions in the declaration of dividends, which certainly reduces the value of the business [9].

The impact of the mentioned factors above greatly complicates the determination of the optimal capital. In addition, the attraction of financial resources from different sources has organizational, legal, macroeconomic and investment restrictions. There is organizational and legal restriction that include legally binding requirements to the value and the order of formation of certain elements of equity and debt, as well as control over the management of the company by the owners.

3 Methodology

In order to invent the model for emerging market companies, we adopted original A. Damodaran's model for determining the optimal capital structure to the emerging markets, with the Russian market in particular. We made several adjustments: for example, if Damodaran's model calculates the cost of equity using the CAPM model, in our calculations, we used the dividend approach. Its application was due to the fact that equity valuation is oriented towards the participants of the stock market, for whom the size of the dividend payment is fundamentally important as an indicator of the company's financial stability. As equity value, we used a specially designed dividend rate corresponding to the current position of the company, which was extrapolated to other ratios of debt and equity. This new model allows objectively evaluate the capital structure of the company, the cost of equity, efficiency of debt use, as well as to develop the parameters of the optimal capital structure of the company in the future.

As a basis for the approbation of the proposed complex dynamic theory approach to the assessment of the optimal capital structure, we selected MTS (Mobile Telesystems, MTS JSC) public joint-stock company, which occupies a leading position in the Russian economy and is characterized as an active participant in the stock market. The issuer which borrows capital on both domestic and international capital markets.

In 2014, the volume of Eurobonds of MTS JSC on the international markets were estimated at 106.477 billion rub., 22.7% of which were short-term securities. Assets in a foreign currency account for about 58% among the company's bonds. At the same time, the largest borrowing amounts from the issue of foreign currency bonds have maturities in 2020-2023, therefore in case of an unfavourable economic situation, the interest paid on these bonds will be high for a long period of time.

The company's problems should also include the downgrade of the credit rating by the three international rating agencies in early 2015. For a company, this means an increase of floating LIBOR. Another issue of MTS JSC is the presence of loans and bonds in foreign currency. Let us have a closer look at the structure of the company's commercial loans. As of June 30, 2014, approximately 25% (43.2 billion rub.) of MTS's loans were in foreign currency MTS. Given the current economic instability, such funds entail high equity risks.

In addition to these external risks, the company also possesses internal ones.

Table 1 shows the main performance indicators of the company for 2010-2014 interest rates on, i.e. the cost of the use of foreign debt in conjunction with the collapse of the ruble significantly increases, making the debt much less profitable. In this case, in addition to interest rate risk, there is also a currency risk related to the high volatility of the ruble against the dollar and the euro.

In 2014, there was a significant decrease in the growth rate of revenue from sales of services, as well as total revenue of the company. The growth rate of operating profit also decreased sharply compared to 2012 and 2013 and is only at 0.58%.

Net profit decreased by 35% compared to 2013, and the growth rate of this indicator is at its lowest in the 4 years under consideration. ARPU (average revenue per user) also shows a very low growth rate and remains at the level of 2013. The same can be said about MOU (minutes of use). In 2014, the average annual asset value of

Table 1. Performance indicators of MTS JSC for 2010-2014.

Indicator	2010	2011	2012	2013	2014
Sales from the communication services	321 589	322 546	349 338	371 950	381 822
Change in %		0.30	8.31	6.47	2.65
Sales from subscriber equipment	21 532	26 025	28 902	26 493	28 936
Change in %		20.87	11.05	−8.34	9.22
Total sales	343 121	348 571	378 240	398 443	410 758
Change in %		1.59	8.51	5.34	3.09
Operating profit	82 996	80 296	93 793	101 758	102 349
Change in %		−3.25	16.81	8.49	0.58
Net profit	46 969	45 939	30 612	80 787	52 393
Change in %		−2.19	−33.36	163.91	−35.15
Average reveue per user	252.8	272.7	297.1	338.6	339.1
Change in %		7.87	8.95	13.97	0.15

Source: Drawn up based on the reports of MTS JSC.

Table 2. Calculation of financial leverage of MTS JSC for 2013-2014.

(mln rub.)	2014	2013
Average annual asset value	539 697	486 844
Net profit	52 393	80 787
Average interest rate on liabilities	7.92%	7.34%
ROA	9.71%	16.59%
Income tax rate, t	20%	20%
Financial leverage	2.29	2.57
DFL	3.3%	19.0%

Source: Calculated based on the information of MTS JSC

MTS JSC increased by 10.85% (Table 2), while the net profit decreased by 35%. Return on assets was at 9.71% in 2014, down by 6.9% compared to 2013. The average cost of debt was at 7.92% in 2014, 7.34% - in 2013.

Performance analysis of MTS JSC has shown that the financial policy of the company has distinct risks. They include currency risk, credit risk, unstable economic situation, a lower rate of revenue growth, as well as the downgrade of the credit rating of both the company and the Russian Federation. Each of the mentioned risk affects the capital cost and financial stability of the company. In order to evaluate the effectiveness of the use of debt by the company, let us calculate the degree of financial leverage.

The financial leverage equal to debt-to-equity ratio was at 2.29 in 2014 and 2.57 in 2013. The degree of financial leverage was calculated using the formula (1):

$$\mathrm{DFL} = (1 - \mathrm{t}) * (\mathrm{ROA} - \mathrm{R}) * \left(\frac{\mathrm{D}}{\mathrm{E}}\right) \tag{1}$$

where DFL is for the degree of financial leverage;
T is for income tax rate;
ROA is for return on assets;
R is for rate on the liabilities of the company;
D is for debt;
E is for equity.

According to our calculations, the degree of financial leverage in 2014 was at 3.3%, which demonstrates the increasing return on equity by this value through the use of borrowed funds. Compared to 2013, this figure is down by 15.7%, which is primarily associated with a significant decrease in net income and an increase in the asset value, which drove the decrease in the return on assets. One of the main indicators to be considered when determining the structure of the capital is the cost of capital (Table 3).

Table 3. Cost of capital of MTS JSC for 2013-2014.

	2014		
mln rub.	Shares	Long-term bonds	Long-term loans
Volume of source	520,000	83,700	157,084
Weight	68.35%	11.00%	20.65%
Cost	9.60%	7.55%	8.14%
Tax discount		20.00%	20.00%
Total	6.56%	0.66%	1.34%
WACC	8.57%		
	2013		
Volume of source	587,100	85,282	108,792
Weight	75.16%	10.92%	13.93%
Cost	6.80%	7.88%	6.85%
Tax discount		20.00%	20.00%
Total	4.09%	0.69%	0.76%
WACC	5.54%		

Source: Calculated based on the information on MTS JSC.

Since the capital structure is defined as the long-term debt-to-equity ratio, short-term liabilities were not taken into account when calculating the cost of capital. It should be noted that when calculating the cost of equity, capitalization of the company was considered as the volume of source, rather than the net book value. The dividend approach was used to determine the cost of capital (2), i.e.

$$\text{Cost of equity} = \text{Paid dividends}/\text{Capitalization} \tag{2}$$

The cost of debt was calculated as the weighted average interest rate based on the interest rates on loans and bonds indicated in the reports of the company.

According to the data presented in Table 3, the cost of equity in 2014 was at 9.6%. This indicator makes equity the most expensive source of funding. It is noteworthy that in 2013, the capital attracted through shares was the cheapest source with its cost at 6.8%. A 2.8% increase in the cost is due to the impact of financial risks, as well as to the increase in debt share in a capital structure of the company to 31.65% from 24.85%. Since the increase in debt capital entails a higher risk, investors demand from the company higher yields on investments, which increases the cost of equity.

In 2014, the cost of debt attracted through bonds was at 7.55%, down by 0.33% compared to 2013 and becoming the cheapest source of capital. There were no significant changes in the structure of debt capital due to the fact that in 2014 the company did not issue new bonds, and only repaid the old ones. However, it should be noted that the lack of a significant decrease in the volume of funds attracts by this source is due to the revaluation of previously issued Eurobonds denominated in US dollars, which is associated with a significant change in the ruble exchange rate.

For MTS JSC the cost of attracting long-term loans in 2014 was at 8.14%, which is 1.25% higher than in 2013. As opposed to the situation with the bonds, in this case the rate increase is associated not only with an increase in the volume of loans in foreign currency, which amounted to 12, 22 billion rubles, but also with the receipt of an additional credit from Sberbank in the amount of 45 billion rubles at the rate increased by more than 2% compared to 2013.

The weighted average cost of capital of the company amounted to 8.57% in 2014, which is 3.03% more than in 2013. In one year the cost of capital rose more than one and a half times, it can be concluded that the company's capital cost grew significantly in 2014. Such an increase in WACC cost the company over 23 bln rubles.

In order to determine the optimal structure of the company, as well as to identify possible ways to overcome the riskiness of the investment policy of the company, let us simulate the value of the company and its capital in different proportions of equity and debt (Bambi, 2015) The initial capital structure is as follows (Table 4).

Table 4. Initial indicators of MTS JSC for 2014

D/(D+E)	31.21%
Cost of equity	9.60%
Cost of debt after tax	6.97%
WACC	8.78%
Value of company	700,563 rub.

Source: Calculated based on the information of MTS JSC.

The main criteria for determining the optimal capital structure are:

- Acceptable levels of return and risk in the company's activity;
- Minimization of the weighted average cost of capital of the company;
- Maximization of the market value of the company.

The optimization process involves the establishment of a target capital structure which determines the level of financial risk of the company in the condition of the maximum value of the company. We calculated the value of the company as the forecasted free cash flows discounted by the weighted average cost of capital (Table 5).

Table 5. Model of the optimal capital structure on the changes of D/E of MTS JSC, mln rub.

D/(D+E)	0.00%	10.00%	20.00%	30.00%	40.00%
D/E	0.00%	11.11%	25.00%	42.86%	66.67%
Debt amount	0,	77,182	154,364	231,547	308,729
Cost of equity	8.50%	8.80%	9.10%	9.60%	10.42%
EBITDA	159,332	159,332	159,332	159,332	159,332
EBIT	84,622	84,622	84,622	84,622	84,622
Interest rate for debt	0	5,714	11,824	20,168	31,240
Taxable profit	84,622	78,908	72,798	64,454	53,382
Tax amount	16,924	15,782	14,560	12,891	10,676
Net profit	67,698	63,126	58,239	51,563	42,705
Cost of debt before tax	7.25%	7.40%	7.66%	8.71%	10.12%
Tax rate	20.00%	20.00%	20.00%	20.00%	20.00%
D/(D+E)	0.00%	10.00%	20.00%	30.00%	40.00%
D/E	0.00%	11.11%	25.00%	42.86%	66.67%
Debt amount	0,	77,182	154,364	231,547	308,729
Cost of equity	8.60%	8.85%	9.10%	9.60%	10.42%
Cost of debt	5.80%	5.92%	6.13%	6.97%	8.10%
WACC	8.60%	8.56%	8.51%	8.81%	9.49%
Value of company	709,606	711,803	714,482	698,979	666,603
D/(D+E)	50.00%	60.00%	70.00%	80.00%	90.00%
D/E	100.00%	150.00%	233.33%	400.00%	900.00%
Debt amount	385,911	463,093	540,275	617,458	694,640
Cost of equity	11.58%	13.47%	16.76%	23.43%	43.08%
EBITDA	159,332	159,332	159,332	159,332	159,332
EBIT	84,622	84,622	84,622	84,622	84,622
Interest rate for debt	41,027	49,233	60,207	75,135	84,526
Taxable profit	43,595	35,389	24,415	9,487	96
Tax amount	8,719	7,078	4,883	1,897	19
Net profit	34,876	28,311	19,532	7,590	76
Cost of debt before tax	10.63%	10.63%	11.14%	12.17%	12.17%
Tax rate	20.00%	20.00%	20.00%	20.00%	20.00%
D/(D+E)	50.00%	60.00%	70.00%	80.00%	90.00%
D/E	100.	150.00%	233.33%	400.00%	900.00%
Debt amount	385,911	463,093	540,275	617,458	694,640
Cost of equity	11.58%	13.47%	16.76%	23.43%	43.08%
Cost of debt	8.51%	8.51%	8.91%	9.73%	9.73%
WACC	10.04%	10.49%	11.27%	12.47%	13.07%
Value of company	642,647	624,374	594,883	554,404	536,394

Table 5 also presents a model for determining the optimal capital structure of MTS JSC. It calculates the approximate cost of the organization and the weighted average capital cost for different ratios of debt and equity capital. A. Damodaran's model for determining the optimal capital structure was used as the basis for the calculations presented in the table. Unlike the original model, where the equity capital cost was calculated using the CAPM model, we used the dividend approach in our calculations. As the equity cost at 30% of debt, the previously calculated rate of 9.6% was used, which corresponds to the current position of the company. This rate was extrapolated to other debt-and-equity ratios.

The cost of debt was calculated similarly, but in this case, there is an issue of interest rates on loans and bonds not reflecting the current situation, as market rates are currently higher than those indicated on the balance sheet of MTS JSC. This is due to the fact that the major rate increase occurred after a significant drop of the ruble in December 2014, i.e., as of December 31, the company has not yet had time to attract debt at a higher cost. In addition, in January-February 2015, both the credit rating of MTS JSC's bonds and the sovereign rating of the Russian Federation were downgraded, which, naturally, is also not reflected on the balance sheet. In this regard, we used the weighted average cost of debt as of the beginning of December 2014, which was increased in proportion to the risk-free rate (long-term federal bonds) from 01.12.2014 to 04.14.2015. The result amounted to 8.71%, which figure was extrapolated to the rest of the capital ratios as per the proportions of the model.

Thus, the model built in accordance with the indicated assumptions demonstrated the following directions for the improvement of MTS JSC financial strategy:

- Selection of the optimal debt-to-equity ratio is determined by the maximum value of the company, which is calculated as its market capitalization and the amount of debt;
- The achievement of the optimal capital structure of MTS JSC requires decreasing the debt down to 20% of the capital, which provides with the given index the greatest value of the company at the lowest capital cost;
- Optimization of the capital structure will increase the value of the company by 15.5 billion rubles and reduce the weighted average capital cost by 0.3%.
- Reduction of the company's risks can be achieved by increasing the share of equity, which will improve the company's credit rating that has an impact on the cost of debt;
- In order to achieve the optimal capital structure in accordance with the simulation results, the value of the equity capital must be reduced by 0.5%, and debt capital by 0.84%.

4 Conclusion

Thus, the modelling of the optimal debt-to-equity ratio of the company can be carried out using a variety of economic models, each allowing to trace the structure and identify the relationship between the main indicators of the financial state of the organization, as well as to calculate the cost of all types of capital. The authors chose

A. Damodaran's method of modelling, which allows determining the optimal debt-to-equity ratio of the company in the most accurate way.

At the same time, instead of the method for calculating the cost of equity using the CAPM model we used the dividend-based approach based on a specially calculated dividend rate corresponding to the current position of the company, which was extrapolated to other ratios of debt and equity. The application of the dividend approach allowed us to reflect the impact of risks on the price of equity capital. This formed the rationale for concluding that in today's economic environment in the emerging market, attracting long-term financing through debt is less advantageous than the use of equity capital or short-term loans.

The approbation of this approach through the example of the risk management at MTS JSC identified several ways to solve the investment problems, aimed at reducing the debt share in the capital structure, which in relation to the current period may be the main direction of minimizing credit, interest rate, and currency risks. For the MTS company, the implementation lies in the transition to the domestic debt markets and increased use of short-term financial instruments. This model of the optimal capital structure may be applicable to other joint stock companies which have entered the securities market, especially in emerging markets. It allows for the companies to support a balance between the benefits of a debt and its expenses, increase the value of the corporation and also to provide financial independence and stability.

References

1. Bambi, M.: Time-to-build and the capital structure. Econ. Lett. **137**, 222–225 (2015)
2. Damodaran's, A.: Website. Available from Internet. http://pages.stern.nyu.edu/~adamodar/. Access date: Nov 2016
3. Drobetz, W., Schilling, D.C., Schroeder, H.: Heterogeneity in the speed of capital structure adjustment across countries and over the business cycle. Eur. Finan. Manag. **21**(5), 936–973 (2015)
4. Fischer, E.O., Heinkel, R., Zechner, J.: Dynamic capital structure choice: theory and tests. J. Finan. **44**, 19–40 (1989)
5. Frank, M.Z., Goyal, K.G.: Tradeoff and pecking order theories of debt, Working Paper, University of British Columbia (2005)
6. Graham, J.R., Leary, M.T., Roberts, M.R.: A century of capital structure: the leveraging of corporate America. J. Finan. Econ. **118**(3), 658–683 (2015)
7. Harris, M., Raviv, A.: Capital structure and information role of debt. J. Finan. **45**, 321–349 (1990)
8. Jensen, M.: Agency: costs of free cash flow, corporate finance and takeovers. Am. Econ. Rev. **76**, 323–329 (1986)
9. Pinegar, M., Wilbricht, L.: What managers think of capital structure theory: a survey. Finan. Manag. **18**, 82–91 (1989)
10. Serrasqueiro, Z., Macas Nunes, P., da Rocha Armada, M.: Capital structure decisions: old issues, new insights from high-tech small- and medium-sized enterprises. Eur. J. Finan. **22** (1), 59–79 (2016)

Specification of Constraints in a System-of-Systems Configuration

Dariusz Caban(✉) and Tomasz Walkowiak

Faculty of Electronics, Wrocław University of Science and Technology,
Wybrzeże Wyspiańskiego 27, 50-320 Wrocław, Poland
{dariusz.caban,tomasz.walkowiak}@pwr.edu.pl

Abstract. A class of Systems-of-Systems (SoS) is considered, where systems are hierarchically composed of subsystems. The structure of the system changes during its lifetime, i.e. component subsystems are moved to other parents. Each system has its configurable parameters. When the configuration changes, it may lead to conflicts in the configuration of its components. There are constraints on component systems configurations that are not limited to the systems, or even to their ancestors in the hierarchy. A domain specific language is proposed to describe constraints in the SoS. It consists of a list of assertions that the SoS configuration must meet. Each assertion is a logical expression that is scoped to a specific subset of component systems.

Keywords: Systems-of-Systems · Configuration · Constraint

1 Introduction

A class of Systems-of-Systems (SoS) [5, 6] is considered, where systems are composed of subsystems. The composition is hierarchically oriented; each component system has its parent system (except for the root SoS), which can be a component of another. The structure of the system may change, i.e. component subsystems can be moved to other parents during the system lifetime. Each system is configurable (has configurable parameters), regardless where in the hierarchy it is situated. The SoS configuration is a composition of its hierarchy of components (composition tree) and the configuration of each of them.

The techniques used for changing system configuration are out of scope for these considerations – the SoS may be self-organizing, manually reconfigured or reconfigured [3] according to a fixed strategy. The issue is that when its configuration changes, it may lead to conflicts in the configuration of its components. There are constraints on component systems configurations that are not internal to the systems, or even to their ancestors in the hierarchy.

The system configuration is described in terms of its parameters (their values). We do not assume any particular format of the system configuration, except that it is tree structured (a few other assumptions presented in Sect. 2). The configuration may be expressed in XML, JSON or any proprietary format describing a structured set of objects of given types. For clarity, we will refer to XML tools for checking the

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_9

constrains, which are more widespread and mature then JSON ones (for example JSON Schema[1]). There is no problem in transforming documents from JSON to XML or serializing memory objects to XML documents, thus this does not limit the applicability of the proposed solutions.

There is no standard for describing constraints in SoS. XML Schemas[2] are adequate for describing constraints within component systems but they are very restrictive when attempting to extend them (using XPaths[3]) to the dependencies between components. Some research papers address general problems of analyzing XML functional dependencies and multivalued dependencies. The main approaches are tree tuple-based [2] and Path-based [7]. However, these works deal with a general XML structure and do not propose any working tools. In the paper we restrict the considerations to configurations of hierarchically composed systems of systems, i.e. a sub model of XML.

A similar problem is discussed in [1, 4], using other formats and types of configuration files.

To describe constraints in the SoS we propose a domain specific language (Parameter Description Language PDL). It consists of a list of assertions that the SoS configuration must meet. Each assertion is a logical expression that is scoped to a specific subset of component systems (based on hierarchy and/or parameter values).

The paper is structured as follows. In Sect. 2, the SoS structure, configuration and types of required constraints are analyzed. In Sect. 3, the proposed PDL language is described.

2 Hierarchically Composed Systems-of-Systems

The considered class of systems encompasses situations, where a system is composed of a number of subsystems. Each subsystem can in turn be composed of sub-subsystems. This leads to a hierarchical composition tree of components. A huge number of systems, software and hardware based, is structured in this manner.

Similar components can occur multiple times in the same system. In the paper, we will be using the terms: objects (single instances of components) and classes (denoting the type of similar components). This corresponds with the object-oriented programming paradigm used when developing the systems.

The configuration of the systems changes during its lifetime, i.e. both the configurations of objects and their hierarchical compositions may change. It should be noted that the changes are assumed to be infrequent in terms of the systems operation – this differentiates the system configuration from the system state, which changes all the time.

[1] http://json-schema.org/.

[2] https://www.w3.org/TR/xmlschema11-1/.

[3] https://www.w3.org/TR/xpath-31/.

2.1 Configuration of SoS

The configuration of a system/subsystem/component is described by a set of its parameters, i.e. configurable variables. The names, number and type of parameters is specific to the type of component being described. So, each **class** has a specific set of parameters. These parameters may be optional (may be omitted in a specific object), may be obligatory (at least one parameter must occur in the configuration) or may have multiple instances. The class specifies the names of the parameters, whereas specific objects are characterized by their values (and number of occurrences).

The configuration of an object is specified by the values of its parameters, but also by the configurations of all the subsystems it is composed of. Thus, if a component is moved from one subsystem to another, both their configurations change, even if all the parameter values remain the same. Reconfiguration occurs if any parameter values are modified, components are moved in the composition tree, components are added or removed from the system, or components are replaced (e.g. when some parts of the system are upgraded).

2.2 Configuration Constraints

Usually, configuration parameters have restrictions placed on their values. When reconfiguring a system, it is quite likely that some of these restrictions might be violated. A well planned reconfiguration strategy must ensure that a system after reconfiguration does not infringe any of the restrictions, i.e. besides the desired changes in the system, some other must be performed so the system does not violate any constraints. A formal specification of the constraints is essential to ensure early identification of invalid reconfigurations.

The configuration constraints can be very varied. Following, there is a discussion of the most common types of constraints and how they can be handled.

- ***Constraints on the number, values and types of parameters***

These are the most common types of constraints. Examples of these constraints include range checks, e.g. a parameter must be in min, max range. They can limit the type of parameter values, e.g. to integers. Also, the number of parameters of a specific name may be constrained to one or more (a mandatory parameter) or to a specific number or range.

These types of constraints are very well handled in case of XML documents by XML Schemas (and in a limited way also by DTD document type declarations). There is no need to develop alternate approaches if the constraints are limited to types, values and occurrence of single parameter.

- ***Constraints on subsystems composition tree***

These constraints restrict the classes of objects that can be the children of a specific parent class. Also, restrictions are placed on the number of child objects.

These constraints can be handled by XML Schema too.

- ***Constraints between parameters of objects in the same composition tree branch***

The examples of such constraints are the requirements that some parameters are smaller or larger than some others in the same object. Often the constraint might not be limited to a single object, but to all the children of a parent object: the limiting value (e.g. max frequency) set in the parent object may require that the corresponding parameter (e.g. operating frequency) in all its children be limited to it.

Even though this type of constraints looks very simple, it cannot be directly enforced with XML Schema. It is fairly simple to express with XPaths and enforce by XSLT[4] processing, but this is hardly user readable. This type of constraints would hardly justify the introduction of a domain specific language, though.

- ***Constraints between parameters of objects related by some other parameter***

The simplest possible constraint of this type is the requirement that all objects of a specific type, having the same value of one parameter, must also have the same value of another parameter, e.g. all network interfaces connected to the same subnetwork must have the bandwidth set to the same value.

A more complex example of this type of constraints is based on cumulative values of a parameter, e.g. the cumulated sum of the power consumed by objects (one parameter) connected to the same power supply(second parameter) cannot exceed the max power of the supply (third parameter).

These types of constraints cannot be handled by the XML Schema. They can be enforced by using some XSLT code, but the problem with this approach is that formulation of the constraint is completely lost. It is almost impossible to deduce the constraint that is being implemented by the XSLT program. While this is a feasible approach for validating the SoS configuration, it is useless to deduce the required configuration changes if the validation fails.

These remarks are also true for the other types of constraints discussed hereafter.

- ***Constraints on parameter uniqueness***

This type of constraint is very frequent in the systems of systems. A recurrent example of such a constraint is the requirement that the same value (e.g. IP net address) is not used twice anywhere in the system. It is very difficult to express in any language that does not literally provide constructs for this purpose.

Actually, the uniqueness requirement becomes more complex to interpret when one considers the possibility of multiple occurrences of the same parameter in an object. Then, the parameter of an object may be considered to have a set of values. Does uniqueness refer in this case just to the values in a single object or to all the objects of the same class? Or maybe, it is required only that the sets of the parameter values must be unique? There must be a way to distinguish

- ***Constraints between parameters of objects fulfilling some other constraint***

A constraint can either be met or be violated. Thus, it can be considered as a specific logical expression. This can be used as a building block of more complex constraints,

[4] https://www.w3.org/TR/xslt20/.

having other constraints embedded in it. The idea is to take all the objects that meet one constraints and require that they also meet another one.

Of course, when using embedded constraints, it is necessary to differentiate the references to parameters between the two constraints (since they may refer to parameters in different instances of the same class object).

3 Domain Specific Language

Parameter Description Language (PDL) was developed for the purpose of defining constraints placed on configuration parameters. It is a domain specific language targeting this field of application. It is targeted to enhance, not to replace XML Schema. Thus, configurations are required to meet a specific schema, which is used to define classes of objects and restrictions on the names and types of parameters that can occur in them. The schema is also assumed to define the permissible values of parameters and permissible hierarchical relationships. PDL is built on top of the schema, to define additional restrictions.

The language references classes, objects and parameter instances described in the schema. These are the variables of the PDL description. In the examples, we use the convention that classes are named with capital letters only, while object parameters have lowercase letters only. It is also possible to use literal and numeric constants, which are denoted similarly to other programming languages.

Assertions form the main functionality of the PDL language. They define the required dependencies between the values of parameters in a configuration. The list of assertions does not depend on the order of occurrence – a configuration is conforming if it meets all the constraints.

Whenever an assertion is described, it must contain information on the objects it should be applied to. This is called the scope of an assertion. Then, it contains a logical expression that evaluates either to true or false when applied to the configuration parameters. In some cases, assertion can be evaluated neither as true nor as false, if it references parameters that are undefined (do not occur) in a referenced object. For this reason, the validator operates on 3-valued logic: true, false and undefined.

If the expression is false, then the configuration fails the assertion. If it is true, then the constraint is met. It is not clear how to interpret an assertion that is undefined. For this reason, we define two validation policies. In case of strict policy, the configuration must meet all the constraints (undefined means that the configuration does not conform to the requirements). Arguably, if any of the parameters in an assertion does not occur, then its relationship to other parameters is immaterial. If this is the philosophy underlying the SoS configuration, then the relaxed validation policy is more adequate – only constraints that are not met (evaluate to false) signify a non-conforming configuration.

The most common form of assertion statement is as:

`SCOPE` **have** `logexpr`

There are various formats for describing the scope and logexpr expressions, e.g.:

```
SYSTEMX have holdAllowed = 'false' unless ats = 'Extended'
```

The strength of PDL lies in its flexibility in defining the scopes of assertions. The simplest declaration is to use class scope, which indicates that the assertion should apply to all the objects of the named class that occur in the analyzed configuration. In the above example, all objects of class SYSTEMX must conform to the requirement.

If the scope is empty, i.e. there are no objects of the specified class in the configuration, then the assertion logexpr is not applied to it. Essentially, this means that the configuration meets this constraint.

The class scope can be limited to a subset of managed objects of the specified class. The most common method of reducing the scope is to define a logical expression that the object parameters must meet to be taken into account: CLASS(logexpr). The assertion is applied only to a subset of the objects that comply with the logexpr, i.e. for which the expression evaluates to true. If the expression is false or undefined, the assertion is not applied to the corresponding object. This means that objects excluded from the scope will not generate errors during validation of this assertion.

The class scope is used when an assertion is applied to all instances of a class, or to a single subset of them. It is not possible to express in this way that the set of class instances has to be split into disjoint subsets and then the expression be applied to each subset separately. The scope in the form of "set of subsets" is used to address this situation.

The set of class instances can be split into subsets in two ways. PDL lets us classify the instances into different subsets on the basis of the value of one or more of its class parameters. The multiple parameters of the same name cannot be used for this to work. Then, the PDL allows CLASS(parameter) notation. When an assertion is defined in the "set of subsets" scope, then it is applied separately to each subset. If the assertion fails on any of the subsets, it fails on the whole scope as well. Lets's consider a simple example. What is the difference in interpreting the following two PDL assertions?

```
SYSW(par1) have par2 identical
SYSW have par3 unique
```

In the first case, all objects of SYSW must have the same value of par2 if they have the same value of par1. In the second case, all the par3 values are different.

A more common way to split the scope into disjoint subsets is to use the hierarchy of the composition tree. The notation used in this case is CLASS/CLASS. The left class corresponds to the parent and the right one to the child. The right-most class defines the scope of the expression. It is split into subsets that are the child objects of the same parent object. Obviously, decomposition is applied on the basis of common parent object (instance, not class). This notion can be extended to any number of parent classes, i.e. notation SYS1/SYS2/SYS3 is interpreted as: class SYS3 is split into subsets having the same parent (SYS2 instance) and grandparent (SYS1 instance).

The class scopes, scope reduction, parameter decomposition and hierarchical decomposition may all be combined to define very complex scopes. In our opinion, this provides unique flexibility when defining configuration constraints. This is complemented by a number of techniques that can be used to define dependencies between the parameters in the scoped objects.

The simplest constraint is just a logical expression. There is nothing noteworthy in this, except for the interpretation of parameter references. Parameter value may be undefined if the parameter does not occur in an object. The reference may have a single value if there is just a single parameter of the specified name. If a parameter occurs multiple times, then the reference to its name yields a set of values. Logical expressions in PDL produce true/false/undefined values according to generally accepted 3-value logic. Sometimes they produce errors if applied to sets.

A very nice feature of PDL is that it operates in conjunction with the schema. The schema provides information, which parameters are mandatory and which cannot occur multiple times. On this basis, PDL constraints can be semantically analyzed during compilation – this helps identify the semantically incorrect rules when they are defined and not when applying them against a configuration.

The exist/not exist assertions complement those defined by a logical relationship. They test if the defined scope contains any objects. If the scope is decomposed into subsets, then the assertion is applied separately to each subset of the scope objects, e.g.:

```
SYSX(par1 = 'true')/SYSY exist
```

is interpreted as the requirement that every SYSX object with par1 set to true, must have at least one associated child of type SYSY.

The assertions "unique" and "identical" test that all the scope objects have a specific parameter value unique or identical, correspondingly. In case of hierarchically decomposed or grouped scopes, the requirement applies to each scope subset separately. PDL language allows the parameter used in unique/identical assertions to occur multiple times in an object. This is interpreted as requiring the sets of values to be unique (not disjoint). So, if in one object par1 = {1,2} and in another par1 = {1} they are treated as unique even though they share the same value par1 = 1.

Contrary to comparing the whole sets, it is often useful to check if some elements do not occur in more than one object. This is addressed by the list-unique construct. A parameter is "list-unique" if within the scope objects the corresponding sets do not have a common element (are disjoint).

Embedded assertions are constructed by including an assertion within a logical expression, either in scope reduction or in dependency. They can be used everywhere within the PDL language, where a logical value is expected. The embedded assertion has its scope defined independently to the enclosing assertion, though it is usually anchored within the external scope. The anchored embedded scope shares an ancestor object with the enclosing scope. It is denoted by including the embedded assertion within brackets in the external assertion. Within the embedded assertions it is possible to reference parameters both from the embedded and the enclosing scope. To distinguish between these references, the external token is used to indicate references to the enclosing scope parameters. Let's consider the following example:

```
SYSP/SYSA/SYSB have (SYSP/SYSA(bId = external SYSB.id) have
cardinality <= 1) if cardinality > 2
```

This example may be split into two assertions. The embedded assertion is interpreted as the requirement that under common SYSP parent object there can be at most 1 SYSA object with parameter bId equal to the id of every SYSB object:

```
SYSP/SYSA(bId = external SYSB.id) have cardinality <= 1
```

The external assertion (SYSP/SYSA/SYSB **have** ?? **if cardinality** > 2, where ?? indicates the result of embedded expression) enforces that the embedded expression matters only for SYSB objects that are not single children of a SYSA object. This example illustrates the complexity of constraints that can be expressed using assertion embedding and the flexibility of the proposed language.

4 Conclusions

We demonstrated in this paper that there is a need for advanced methods of expressing configuration constraints. While simple configurations can be constrained with schemas, this is not the case if constraints define various interdependencies between parameters. We proposed a domain specific language to solve this deficiency. The language is closely related to configuration schemas, which adds to its applicability. The tools supporting the language have been developed at Wroclaw University of Science and Technology. The language has already been adopted by a software development company for maintaining configuration consistency of their products.

The proposed formal specification of configuration constraints can further be used to introduce sophisticated reconfiguration strategies in the systems of systems. It can also provide a basis for identifying conflicting requirements as well as automatic resolving of conflicts in the target reconfiguration.

References

1. Anderson, P., Herry, H.: A formal semantics for the SmartFrog configuration language. J. Netw. Syst. Manage. **24**(2), 309–345 (2016)
2. Arenas, M., Libkin, L.: A normal form for XML documents. ACM Trans. Database Syst. **29** (1), 195–232 (2004)
3. Caban, D., Walkowiak, T.: Preserving continuity of services exposed to security incidents. In: Proceedings of the Sixth International Conference on Emerging Security Information, Systems and Technologies, SECURWARE, pp. 72–78 (2012)
4. Delaet, T., Anderson, P., Joosen, W.: Managing real-world system configurations with constraints. In: Proceedings of the Seventh International Conference on Networking, pp. 594–601 (2008)
5. Jaradat, R.M., Keating, C.B., Bradley, J.M.: A histogram analysis for system of systems. Int. J. Syst. Syst. Eng. **5**(3), 193–227 (2014)
6. Keating, C.B., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A.A., Safford, R., Peterson, W., Rabadi, G.: System of systems engineering. Eng. Manage. J. **15**(3), 35–44 (2003)
7. Vincent, M.W., Liu, J., Liu, C.: Strong functional dependencies and their application to normal forms in XML. ACM Trans. Database Syst. **29**(3), 445–462 (2004)

A Methodological Framework for Model-Based Self-management of Services and Components in Dependable Cyber-Physical Systems

DeJiu Chen[1(✉)] and Zhonghai Lu[2]

[1] Mechatronics, Machine Design, School of ITM,
KTH Royal Institute of Technology, 100 44 Stockholm, Sweden
chendj@kth.se

[2] Electronics and Embedded Systems, School of ICT,
KTH Royal Institute of Technology, 164 40 Kista, Sweden
zhonghai@kth.se

Abstract. Modern automotive vehicles featuring ADAS (Advanced Driving Assistant Systems) and AD (Autonomous Driving) represent one category of dependable CPS (Cyber-Physical Systems). For such systems, the adaptation of generic purpose COTS (Commercial-Off-The-Shelf) services and components has been advocated in the industry as a necessary means for shortening the innovation loops and enabling efficient product evolution. This will however not be a trivial task due to the system safety- and time-criticality. This calls on one hand for formal specification of systems, and on the other hand for a systematic approach to module design, supervision and adaptions. Accordingly, we propose in this paper a novel method that emphasizes an integration of system models, formal contracts, and embedded services for effective self-management of COTS. The key modeling technologies include the EAST-ADL for formal system description and the A-G contract theory for module specification.

Keywords: Cyber-Physical systems · Commercial-Off-The-Shelf · Dependability · Real-time · EAST-ADL · Contract · Composition

1 Introduction

Embedded system in modern automotive vehicle allows more effective realization of advanced functionalities in a way that is impossible with pure mechanical and electrical solutions. It makes an automotive vehicle CP (Cyber-Physical) in nature by having both physical dynamics and energy flows under control, and the corresponding perception, control and cognitive loops with software and hardware solutions. Currently, the industry shares the view that the embedded system constitutes the most important technology for the advances in sustainability, road safety, and novel traffic solutions [1, 2]. For shortening the innovation loops and enabling efficient product evolution, the adaptation of generic purpose COTS (Commercial-Off-The-Shelf) solutions for the embedded system has been considered as necessary. Such COTS solutions typically

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_10

range from camera, radar and other sensors for traffic perception, to speech recognition and augmented-reality displays, and to wireless and telecommunication services for V2V and V2I connectivity. There are however often big gaps in regard to the conformity of qualities. For example, the expected lifetime for electronics components would be up to 15 years for automotive vehicles, in comparison to a length in 2–5 years for consumer products [3]. Meanwhile, the operational temperature ranges are normally –40 ~ 160°C for automotive vehicles and 0 ~ 40°C for consumer products. Moreover, for the functional safety of automotive embedded systems, a systematic management of the risks of COTS according to the safety standard ISO26262 becomes necessary [4]. The information of concern typically includes not only a specification of its functional behaviors, but also a definition of its assumed failure modes, adopted quality assurance measures, and related mechanisms for fault tolerance and treatment. This is not a trivial task for a separately developed service or component.

This paper presents a novel approach that aims to act as a methodological framework for effective integration and management of separately developed services and components in safety critical CPS. The aim is to promote in general qualified CPS development and evolution across the boundaries of domains and organizations. The rest of this chapter is structured into the following sections: Sect. 2 provides an overview of the proposed methodological framework. Section 3 introduces EAST-ADL, which is the state-of-the-art technology adopted for systems modeling and modularity design. Section 4 presents the contract theory used as the formalism for module specification. Section 5 describes the corresponding run-time services developed for situation awareness and system adaptation. An overview of related technologies is given in Sect. 6. The paper concludes with Sect. 7.

2 The Methodological Framework

The methodology framework emphasizes an integration of the models for system design and the service for post-deployment quality management. It follows the knowledge-in-the-loop paradigm proposed in [5]. Shown in Fig. 1, the framework consists of the following four main steps:

- **Step I – System architecture design**, responsible for the *overall system design* in regard to the definition of system operational environments, the functional and technical constituent units, the expected behaviors and quality constraints, the corresponding verification and validation cases, etc. As the outcome, the design also stipulates the design-space with compositional variants and restrictions.
- **Step II – Partitioning and assertion specification**, responsible for *system modularity design*, which is centric on partitioning the target system into modules and thereby refining the system design for potential external COTS solutions. In effect, the design results in a restructuring of the target system so that some of its constituent units can be developed and managed independently. We specify the constraints with the *module assertions* with the allowed input and output signals, their preferred types and resolutions, execution time and criticalities, etc.

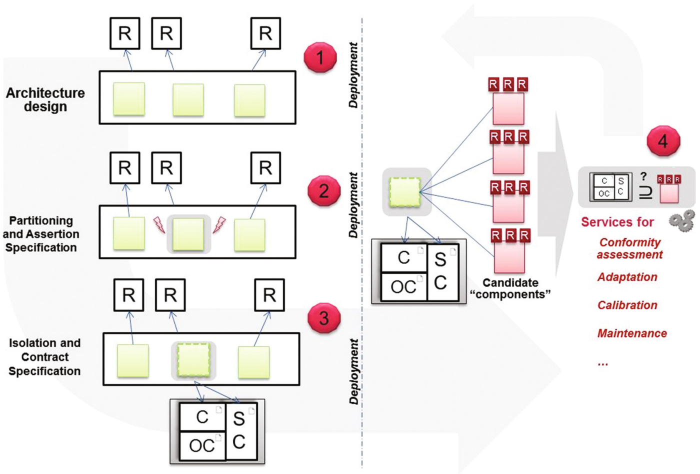


Fig. 1. The overall methodology for a model-based contract and service for self-management. ① Step I – Architecture design; ② Step II – Partitioning and Assertion Specification; ③ Step III – Isolation and Contract Specification; ④ Step IV – Service for self-management. R- Requirement; C- Contract; OC-Operation Contract; SC-Safety Contract.

- **Step III – Isolation and Contract Specification**, responsible for *module interface design* for the given modules. By *isolation*, we refer to the process of configuring the target system to allow each module, as a self-contained system unit, to be isolated from changes or variations in the target system and in the COTS. The results for modules are specified as *contracts*. Each contract formally specifies some assertions for successful COTS integration, on the functional, operational, and safety properties, referred to as Contract (C), Operation Contract (OC) and Safety Contract (SC) respectively in Fig. 1. While the functional properties focus on the input and output signal as well as the functions, the operational properties are related to the nominal operation and the modes for diagnostics and maintenance. The safety properties cover the failure modes, safe states, and the related transitions for error handling and fault treatment.
- **Step IV – Service-based Self-Management of System Services and Components**, responsible for the development and usage of *quality management services* for post-deployment monitoring, assessment and adaptation. These services take the module contracts as formal specifications of the parameters to be monitored, the conditions to be judged, and the adaptations to be conducted.

The realization of this methodology is based on the combination of some state-of-the-art technologies for system modeling, component specification, and self-management. We introduce these technologies by the follow-up sections.

3 EAST-ADL for System and Modularity Design

ADL (Architecture description language) is a modeling technology for structuring and managing the information of a system, relating to the *Step I* and *Step II* introduced previously. The EAST-ADL modeling framework, adopted here, represents a key European initiative towards a standardized description of automotive E/E systems [6, 7]. The EAST-ADL system model, together with the associated requirements and constraints, constitutes a basis for effective but still very flexible modularity design. That is, given the models capturing the system wide interdependencies, a particular modularization task can be driven at any specific abstraction level in accordance with the particular preferences. For example, the modules given by features (Vehicle Level) provide a structuring of the externally visible functionalities of the target system and allow a service-oriented composition of external functions. Meanwhile, module definitions can also be done for software components (Implementation Level) to allow a component-based engineering of software solutions. In both cases, the EAST-ADL system model constitutes a useful means for systematically reasoning about the corresponding module coupling and cohesion, both in regard to the functional interactions (which are given by communication links in the same level of abstraction) and technical implications (which are given by realization links across the abstraction levels). By capturing the associated requirements and constraints of system constituent units, an EAST-ADL model also provides a systematic modeling support for deriving the related module requirements and constraints.

4 A-G Contract for Formal Specification of Modules

Following the contract theory defined in [8, 9], we use formal contracts to specify the expected functional and technical properties of a system module, relating to the *Step III*. Fundamentally, a system module M is defined by its variables, behaviors as well as some related quality constraints as follows

$$M = (\mathbb{V}, \mathbb{F}, \mathbb{Q}) \tag{1}$$

with $\mathbb{V}$ for all variables $\mathbb{V} = \mathcal{U} \cup \mathcal{X} \cup \mathcal{Y} \cup \mathcal{K}$, where $\mathcal{U}$ denotes the input variables, $\mathcal{X}$ the internal variables, $\mathcal{Y}$ the output variables, and $\mathcal{K}$ the configuration variables; $\mathbb{F}$ for all functional behaviors over the variables $\mathbb{F} = [\![\mathcal{F}(\mathcal{U}, \mathcal{Y}, \mathcal{X}, \mathcal{K}) = 0]\!]$; and $\mathbb{Q}$ for all quality constraints over the functional behaviors $\mathbb{Q} = \{\mathbb{P}, \mathbb{R}, \ldots\} = \{[\![\mathcal{P}(\mathbb{F}) = 0]\!], [\![\mathcal{R}(\mathbb{F}) = 0]\!], \ldots\}$. with $\mathbb{P}$ for all performance constraints and $\mathbb{R}$ for all reliability constraints, etc. For successful system integration, all COTS solutions implementing a module implementation need to satisfy the expected module properties given by M, under certain environmental conditions. Such a requirement is formally defined by module contract $\mathcal{C}$ in terms of the pair:

$$\mathcal{C} = (\varepsilon_c, \mathcal{M}_c) \tag{2}$$

with $\mathcal{M}_{\mathrm{c}}$ for all COTS solutions satisfying the M; and ε_{c} for all the legal environmental conditions. A contract inconsistent if $\mathcal{M}_{\mathrm{c}} \neq \emptyset$ and compatible if $\varepsilon_{\mathrm{c}} \neq \emptyset$. A component $\mathrm{M}_{\mathrm{comp}} \vDash \mathcal{C}$ if and only if $\mathrm{M}_{\mathrm{comp}} \in \mathcal{M}_{\mathrm{c}}$; An environment $\mathrm{E} \vDash \mathcal{C}$ if and only if $\mathrm{E} \in \varepsilon_{\mathrm{c}}$.

A-*G* (Assume-Guarantee) contract is a formalism for the description of contract $\mathcal{C}$ without directly referring to the actual COTS solutions $\mathcal{M}_{\mathrm{c}}$. Normally, it is used for behavioral specifications of components, with *A* for the constraints on acceptable behaviors of the environment and *G* for the guarantees in terms of the corresponding component behaviors (as in e.g. [9]). Here, we use the A-G contract formalism in a more generic sense for all constraints that are characterized by logical or technical causalities from some environmental conditions (*A*) to some consequential module properties. Formally, an *A*-*G* contract with logical or technical causality is defined as:

$$A \Rightarrow G \tag{3}$$

with *A* for the constraints on the acceptable environment conditions; and *G* for the corresponding constraints on the properties to be guaranteed by module implementations. There is a logical or technical causality between *A* and *G* (i.e. $A \ \cap M \subseteq G$). Currently, we use STL (Signal temporal logic) [10] and PrSTL (Probabilistic Signal temporal logic) [14] to define the constraints.

Example for the transformation of system constraint to module contract: A vehicle braking system has been introduced in [7] as a case study for EAST-ADL. For the

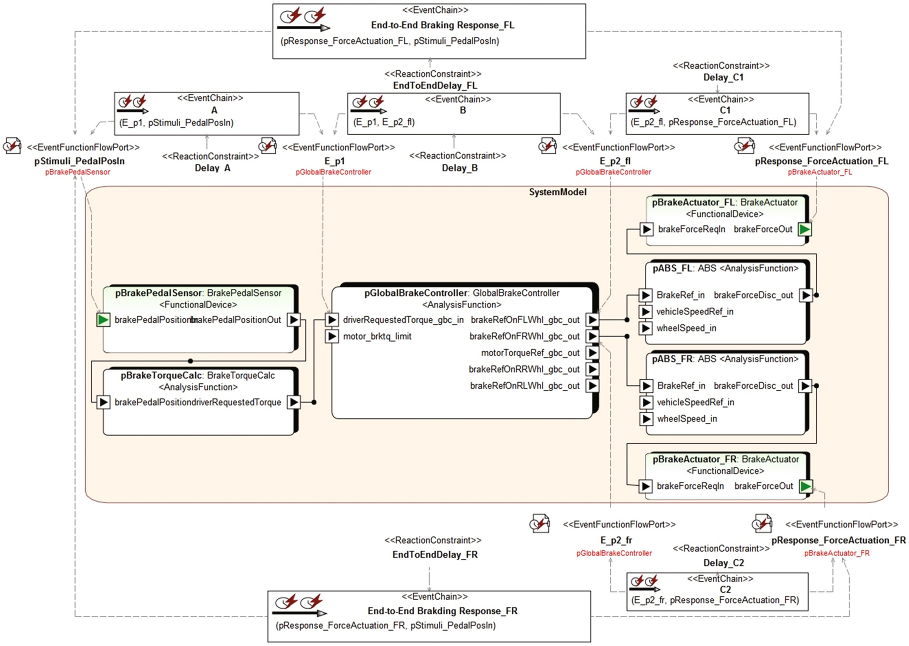
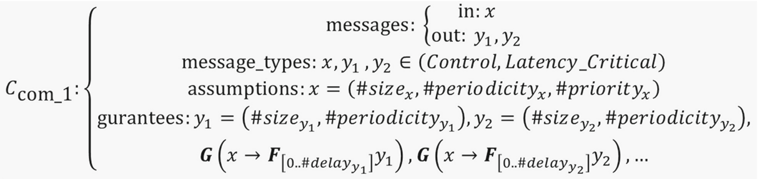


Fig. 2. The architecture model of an automotive vehicle braking system and the timing specification in EAST-ADL (implemented with the DSM Workbench MetaEdit+)

braking control, a brake controller receives the driver braking request and then sends brake force request to ABS controllers. Figure 2 shows an excerpt of the system model (*SystemModel*). The system design is converted to *A-G* contracts for module specification. Suppose the entire braking system will be based on COTS, the *A-G* contract $C_{\text{com\_1}}$, shown in Fig. 3, stipulates the communication and timing assertions.

The timing specification in Fig. 2 augments the system model with information about the executional events and their timing constraints. In the model, an executional event defines the occurrence of data arrival on port (e.g. *pStimuli_PedalPosIn*), and the trigging of execution (e.g. *E_p1*). An event chain binds then these events for synchronization, e.g. to capture the end-to-end timing from sensor input to actuator outputs. In the model, the event chains *End-to-End Braking Response_FL* and *End-to-End Braking Response_FR* capture the timing requirement from driver braking request (*pStimuli_PedalPosIn*) to the brake torque actuation on the FL and FR wheels (*pResponse_ForceActuation_FL*, *pRsponse_ForceActuation _FR*) respectively. For a module based implementation, the contract in Fig. 3 declares the related executional events as contract variables: x for *pStimuli_PedalPosIn*, y_1 for *pResponse_ForceActuation_FL*, y_2 for *pRe sponse_ForceActuation_FR*. All these events are of the types *control* and *latency critical*. Given the assumption x about its *size*, *periodicity* and *priority*, an external solution for the braking system needs not only to guarantee the preferred size and periodicity y_1 and y_2, but also to meet the delays $x \rightarrow F_{[0..\#delay_{y_1}]} y_1$ and $x \rightarrow F_{[0..\#delay_{y_2}]} y_2$ for the torque response on the FL and FR wheels respectively.



$$C_{\text{com\_1}}: \begin{cases} \text{messages: } \begin{cases} \text{in: } x \\ \text{out: } y_1, y_2 \end{cases} \\ \text{message\_types: } x, y_1, y_2 \in (Control, Latency\_Critical) \\ \text{assumptions: } x = (\#size_x, \#periodicity_x, \#priority_x) \\ \text{gurantees: } y_1 = (\#size_{y_1}, \#periodicity_{y_1}), y_2 = (\#size_{y_2}, \#periodicity_{y_2}), \\ \boldsymbol{G}\left(x \rightarrow \boldsymbol{F}_{[0..\#delay_{y_1}]} y_1\right), \boldsymbol{G}\left(x \rightarrow \boldsymbol{F}_{[0..\#delay_{y_2}]} y_2\right), \ldots \end{cases}$$

Fig. 3. The A-G contract for the vehicle braking. Here, we use STL (Signal temporal logic) [10] to denote the constraints, with the operator G for always and F for eventually.

5 Software Services for Self-management of System Services and Components

An embedded system is self-managed if it is able to autonomously understand the actual operational situations and alter its own configurations, behaviors to meet the requirements [11]. As a further support for COTS, relating to the *Step IV,* two embedded software services have been introduced. These services, shown in Fig. 4, allow error detection and fault treatment, while complementing the verification and validation at development time with the post-deployment analysis. These two services are:

- **Monitoring and Assessment Service,** responsible for self-assessment by perceiving the operational conditions of component or system (e.g., the actual input and output conformity, the CPU and memory utilization). Contracts are used for the

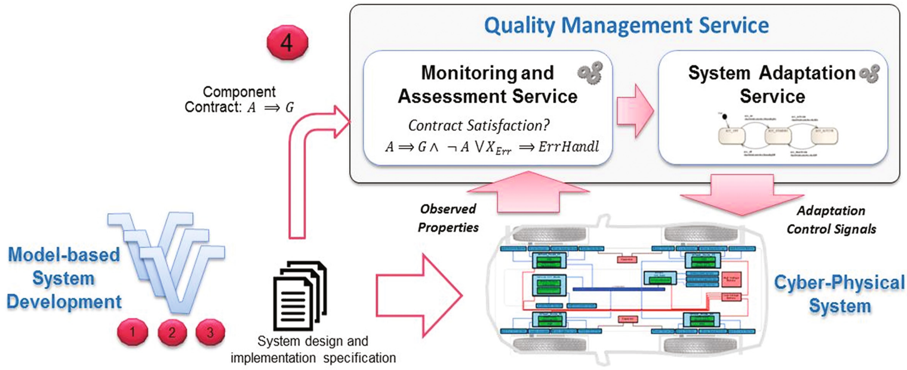


Fig. 4. The architecture of software services for self-management system services and components. (Note, ①②③④ are for the methodological steps as depicted in Fig. 1).

decision-making. For example, for a component, the service monitors the external conditions (A) and internal error states (X_{Err}); it triggers error handling ($ErrHandl$) when necessary, i.e. $A \Rightarrow G \wedge (\neg A \bigvee X_{Err} \Rightarrow ErrHandl)$.

- **System Adaptation Service,** responsible for planning and controlling configuration changes for quality-of-service adaptation and error handing. The decision-making requires built-in knowledge about the configuration variability. For dependability, the ability of inferring the causing factors of anomalies (e.g. based on the minimum-cut sets of fault-tree analysis) plays a key rule for the success. The adaptation is controlled by deterministic or probabilistic state-machines.

6 Related Technologies

This paper presents a methodological framework for a model-based integration and management of COTS in safety critical CPS. There are many modeling frameworks useful for the description and management of embedded systems, such as SysML [12], AADL [13] and EAST-ADL [6, 7]. All these technologies focus on the system development. For dealing with the partially unknown conditions and emergent properties, the specification with abstract goals, control policies and contracts together with the provision of intelligent services for operation supervision and adaptation becomes necessary. For example, a mission goal description based approach to the identification of required capabilities for system operations has been proposed in [15]. A contractual description of component interfaces, considering the imprecision and uncertainty, is given in [16]. For quality assurance and certification, such contractual support however needs to be managed seamlessly along with the system development and componentization. A great many techniques have recently been developed for advanced verification and validation. One promising technique is on-line model checking as explored in [17]. In this approach, the state space is continuously monitored against critical safety properties. Another approach that combines model-checking with self-assessment is learning-based testing (LBT) [18]. New approaches to operational risk assessment also include statistical analysis of incident data using Bayesian theory [19]. In regard to all the above mentioned approaches, our work aims to facilitate the integration and adoptions of technologies.

7 Conclusion

For CPS, the need to integrate COTS solutions, supporting dependable execution and change management, calls for an integrated approach as proposed by this paper. The complexity and safety concern demand both more formal approach to system development and more advanced mechanisms for supervision and quality management. With EAST-ADL for system modeling, our work has been focused on the provision of contract formalism for module specification as well as embedded services for situation awareness and adaptation. Currently, we are looking into how to enable run-time supervision and control through the software stack of services for mixed-criticality communication based on Ethernet and AUTOSAR.

References

1. SAE International, SAE Information Report: (J3016) Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems
2. European Commission: Intelligent transport systems. https://ec.europa.eu/transport/themes/its_en
3. PwC Semiconductor Report: Spotlight on Automotive. PwC, September 2013
4. ISO, ISO 26262 Road vehicles – Functional safety
5. Chen, D., et al.: A Knowledge-in-the-loop approach to integrated safety&security for cooperative system-of-systems. In: IEEE 7th International Conference on Intelligent Computing and Information Systems, ICICIS 2015, Cairo, Egypt, 12–14 December (2015)
6. EAST-ADL. EAST-ADL Domain Model Specification, Version M.2.1.12 (2014). http://www.east-adl.info/
7. Kolagari, R., et al.: Model-based analysis and engineering of automotive architectures with EAST-ADL: revisited. Int. J. Conceptual Struct. Smart Appl. (IJCSSA) **3**(2), 25–70 (2015)
8. Benveniste, A., et.al.: Multiple viewpoint contract-based specification and design. In: Formal Methods for Components and Objects: 6th International Symposium, FMCO, pp. 200–225 (2007)
9. Benveniste, A., et.al.: Contracts for system design. Research Report 8147, Inria, November 2012
10. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: Joint International Conference on Formal Modelling and Analysis of Timed Systems, and Formal Techniques in Real-Time and Fault-Tolerant Systems (FORMATS/FTRTFT 2004) (2004)
11. Anthony, R., et al.: Context-aware adaptation in DySCAS. Electronic Communications of the EASST, vol. 19. European Association *of* Software Science and Technology (EASST) (2009)
12. SysML. OMG Systems Modeling Language (OMG SysML™), OMG
13. AADL Architecture Analysis and Design Language, SEI. Carnegie-Mellon Univ., USA

14. Sadigh, D., Kapoor, A.: Safe Control under Uncertainty with Probabilistic Signal Temporal Logic. Robotics: Science and Systems (RSS), June 2016
15. Silva, E., Batista, T., Oquendo, F.: A mission-oriented approach for designing system-of-systems. In: Proceedings of the 10th System-of-Systems Engineering Conference (SoSE), May 2015
16. Bryans, J., Fitzgerald, J., Payne, R., Miyazawa, A., Kristensen, K.: SysML contracts for systems of systems. In: IEEE Systems of Systems Engineering Conference, June 2014
17. Althoff, M., et al.: Online verification of automated road vehicles using reachability analysis. IEEE Trans. Robot. **30**(4), 903–918 (2014)
18. Meinke, K., et al.: Incremental learning-based testing for reactive systems. In: Proc. Int. Conf. on Tests and Proofs TAP 2011. LNCS, vol. 6706, Springer (2011). IEEE Trans. Robot. **30**(4): 903–918 (2014)
19. Meel, A.: Plant-specific dynamic failure assessment using Bayesian theory. Chem. Eng. Sci. **61**, 7036–7056 (2006)

Maintenance of Wind Turbine Scheduling Based on Output Power Data and Wind Forecast

Guglielmo D'Amico[1], Filippo Petroni[2],
and Robert Adam Sobolewski[3](✉)

[1] Department of Pharmacy, University "G. d'Annunzio" of Chieti-Pescara,
Italy, via dei Vestini 31, 66100 Chieti, Italy
g.damico@unich.it

[2] Department of Business and Economics, University of Cagliari,
Italy, V.le S. Ignazio 17, 09123 Cagliari, Italy
fpetroni@unica.it

[3] Department of Power Engineering, Photonics and Lighting Technology,
Bialystok University of Technology, Wiejska 45D, 15-351 Bialystok, Poland
r.sobolewski@pb.edu.pl

Abstract. Maintenance of a wind turbine is a combination of all technical, administrative and managerial actions intended to retain it in, or restore it to, a state in which the turbine is able to generate power. This paper presents an influence diagram to estimate the expected utility that represents wind turbine energy to be produced given period of time in the future. The conditional probability distribution of a chance node of the diagram is obtained relying on Bayesian networks, whereas the utilities of value node are calculated thanks to the second order semi-Markov chains. The example shows the application of the models in the real case of one wind turbine E48 by Enercon located in northern part of Poland. Both Bayesian network parameters and kernel of semi-Markov chain are derived from real data recorded by SCADA system of the turbine and weather forecast.

Keywords: Maintenance scheduling · Wind energy · Influence diagram · Second order semi-Markov chain · Bayesian networks

1 Introduction

Maintenance of a wind turbine is a combination of all technical, administrative and managerial actions intended to retain it in, or restore it to, a state in which the turbine is able to generate power. There are different maintenance strategies, e.g. preventive and corrective ones. For example, preventive maintenance is dedicated to reducing the probability of failures or the degradation of the turbine's performance. Very often the turbine should be out of service during the maintenance action. Performing the maintenance of the wind turbine under good wind conditions may lead to energy not served. To restrict this energy the maintenance scheduling should take into account the

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_11

prediction of the amount of energy to be produced within the planned period of time of the maintenance action.

The research in the field of wind turbines' maintenance is focused mainly on estimation of the effects of different maintenance strategies and their optimization, to limit the maintenance cost and production losses due to turbine's downtime [1–3]. Concerning production losses in these research works, losses estimation rely on simplified approaches that do not assure the reasonable results, e.g. wind speeds are averaged over 24 h time interval using historical data [1]. These approaches could be updated with the stochastic model intended to make decisions on the best time of the maintenance work that minimizes the wind turbine energy yield because of the turbine being out of service.

This paper presents an influence diagram to estimate the expected utility that represents wind turbine energy to be produced given period of time in the future. The conditional probability distribution of a chance node of the diagram is obtained relying on Bayesian network, whereas the utilities of value node are calculated thanks to the second order semi-Markov chains. The example shows the application of the models in the real case of one wind turbine E48 by Enercon located in northern part of Poland. Both Bayesian network parameters and kernel of semi-Markov chain are derived from real data recorded by SCADA system of the turbine and weather forecast.

2 Maintenance Scheduling as a Decision Problem

Appropriate maintenance scheduling of wind turbine is a decision problem, involving the time of starting and duration of the maintenance activity. To make the best decision that minimizes the energy not supplied because of the disabled wind turbine, the influence diagram can be used. Such a diagram is an effective modeling framework for representation and analysis of the decision making process under uncertainty caused by stochastic nature of the wind turbine energy yield.

An influence diagram can be considered as a Bayesian network augmented with decision variables, utility functions defining the preferences of the decision maker, and precedence ordering specifying the order of decisions and observations [4]. The objective of decision analysis is to identify the decision option that produces the highest expected utility. To identify the decision option with the highest expected utility one needs to compute the utility of each decision alternative. If A is a decision variable with options $a_1, \ldots, a_m$, H is a hypothesis with states $h_1, \ldots, h_n$ and ε is a set of observations in the form of evidence, then one can compute the utility of each outcome of the hypothesis and the expected utility of each action. The expected utility of action a_i is

$$\mathrm{EU}(a_i) = \sum\nolimits_j \mathrm{U}\left(a_i, h_j\right) \cdot \mathrm{P}\left(h_j|\varepsilon\right) \tag{1}$$

where $\mathrm{P}(\cdot)$ represents the belief in H given ε and utility function $\mathrm{U}(\cdot)$ encodes the preferences of the decision maker on a numerical scale.

One shall choose the alternative with the highest expected utility. This is known as the maximum expected utility which amounts to selecting an option a^* such as

$$a^{*} = arg\,\max_{a \in A} \mathrm{EU}(a). \tag{2}$$

In Fig. 1 one can see the diagram we created for the decision making problem of wind turbine scheduling. It contains three types of nodes: decision, chance and value. There are two decision nodes (drawn as rectangles) that represent decision variables, i.e. 'Starting point of maintenance' and 'Maintenance duration'. The options for the first node refer to a discrete moment of time when the maintenance can be started, whereas the options for the second node involve the number of time instants within the maintenance to be taken. There is one chance node (drawn as oval) 'Wind power scenario' that represents hypothesis with states determined by the combination of the wind turbine output power states (total number of states is s) within two discrete moments of time: (i) when the maintenance to be started and (ii) one unit of time preceding the starting point of the maintenance. The number of wind power scenarios depends on the starting point of maintenance activity. The s states represent different partitions of wind turbine output power magnitudes within the power range 0 … P_R (rated power of wind turbine). They can be either observed from real data of wind turbine output power or predicted from the forecast of wind speed and direction. 'Wind power scenario' is a random variable quantified by conditional probability distribution calculated given the option of the decision node 'Starting point of the maintenance'. There is one value node (drawn as a diamond) that represents the 'Expected utility'. This utility can be calculated relying on predictive model of wind turbine output power trajectory given declared options of decision nodes and conditional probability distributions of a chance node. The trajectory represents the process of evolution of power and energy within each unit of time. In this case one shall choose the alternative with the lowest expected utility (opposite to formula (2)) since we look for the decision that would ensure the minimum energy to be lost because of the disability of wind turbine caused by its maintenance activity. The node is quantified by the utility of each of the combination outcomes of the parent nodes.

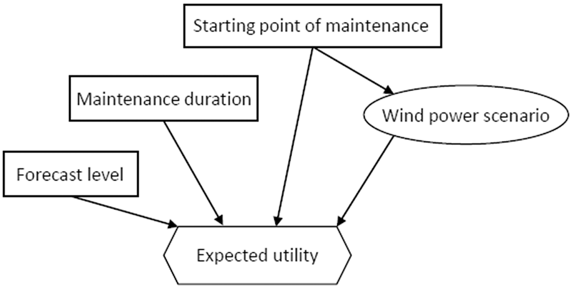


Fig. 1. Influence diagram for decision making problem of wind turbine maintenance scheduling

An arc in Fig. 1 denotes influence, i.e. the fact that the node at the tail of the arc influences the value (or the probability distribution over possible values) of the node at the head of the arc. Some arcs have a causal meaning, e.g. the path from a decision node to a chance node means that the decision will impact the chance node in the sense of changing the conditional probability distribution over its outcomes.

Graphical representation of the wind turbine output power states along the timeline of discrete time units is depicted in Fig. 2. It presents the idea of obtaining: wind power scenario probabilities and output power trajectory needed for calculation of energy yield. Below the timeline one can see the *s* states of output power for each time *t*. Number of states *s* is the same for each *t* and should be optimized regarding the satisfied error of the results to be obtained based on the models described in Sect. 3. The output power states and their probabilities within *t* can be derived relying on either real data recorded by SCADA system of wind turbine (up to $t = 0$) or prediction (from $t = 1$). The state within *t* means that output power occupies this state for one instant of time, e.g. 1 h. The power state within *t* derived thanks to real data occupies this state with probability 1 whereas the power states (1 … *s*) predicted within *t* have conditional probability distribution (joint distribution). The Bayesian networks approach is used to derive wind power (WP) prediction based on wind speed (WS) forecast and wind direction (WD) forecast, within *t*. Both of the meteorological forecasts are sometimes accessible for the wind turbine (or farm) operator. If the forecast of wind speed and direction at the wind turbine site are not accessible one can obtain the wind output power trajectory relying on the real data recorded by SCADA system only. In such case the satisfying trajectory can be obtained for quite a short period of time (around 50 instants of time). One can find details of BN model of WPP in Subsect. 3.1.

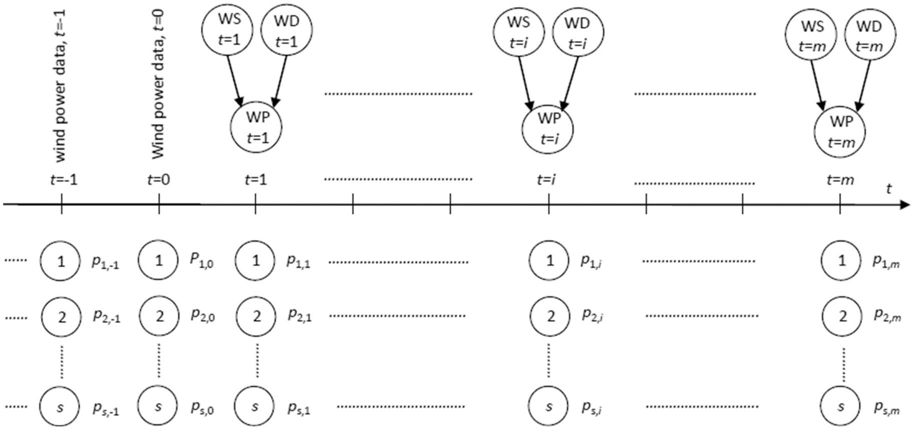


Fig. 2. Graphical representation of the wind turbine output power states (observed and forecasted) along the timeline of discrete time instants. Designations in the text

Wind power scenario (WPS) probability within $t = i$ (let's assume that starting point of maintenance activity is *i*) is a multiplication of the probabilities of two wind output power states – within $t = i$ and $t = i–1$. When $i = 0$ probability of wind power scenario is 1 whereas for $i = 1$ – this probability equals $p_{u,1}$, $u = 1,2,…,s$. When $i > 1$ then the wind power scenario probabilities are derived as follows

$$\mathrm{p}_g^{\mathrm{WPS}}(i-1,i) = \mathrm{p}_{u,i-1} \cdot \mathrm{p}_{u,i} \tag{3}$$

where $g = 1, 2, \ldots, s^2$.

The second order semi-Markov chain is used to derive wind turbine output power trajectory given output power state within both $t = i$ (starting point of maintenance activity) and $t = i - 1$, sojourn time in state within $t = i - 1$, and maintenance duration. Since the number of wind power scenarios for $t = 1$ can be higher than 1 it is needed to find the output power trajectories for all power scenarios. One can find details of the second order semi-Markov chains in Subsect. 3.2. Energy yield is calculated as follows

$$\mathrm{EY}_{i,x,g} = x \cdot \mathrm{P}_{i,g} \tag{4}$$

where x is the number of time instants (maintenance duration) and $\mathrm{P}_{i,g}$ is the wind turbine output power that represents power state within $t = i$.

The power $\mathrm{P}_{i,g}$ can be the middle value of i-th partition of the output power range (0 … P_R). Expected utility given i and x is calculated as follows

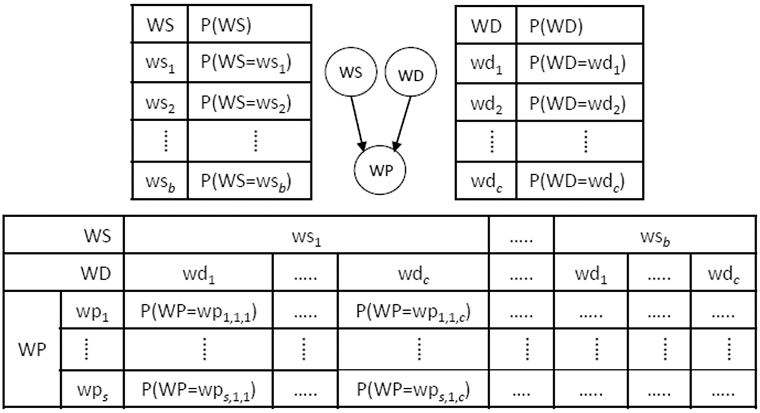
$$\mathrm{EU}_{i,x} = \sum\nolimits_g \mathrm{EY}_{i,x,g} \cdot \mathrm{p}_g^{\mathrm{WPS}}. \tag{5}$$

3 Modeling of Wind Turbine Output Power

3.1 Bayesian Network Model Used for Output Power Reasoning

The Bayesian network model used for wind power prediction is depicted in Fig. 3. One model can be used within $t = 1,\ldots,m$. The elements of this graphical model are nodes, arcs between the nodes, and probability assignments. There are two root nodes WS and WD, and they are parents of one child node WP. Each node represents a random variable with a finite set of mutually exclusive states (a state is a possible value or partition of values, associated with the random variable). The arcs represent direct relevance relationships among variables. It is assumed that all the variables are discrete ones. It means that the continuous real data need to be discretized into some states.

The conditional probability table (CPT) associated with the nodes is provided close to them in Fig. 3. For WP with parents WS and WD there is CPT $\mathrm{P(WP|WS, WD)}$. The CPT of the nodes WS and WD come down to probabilities $\mathrm{P(WS)}$ and $\mathrm{P(WD)}$, unconditional on other nodes in the graph. The random variable WS is of states ws_1, ws_2, …, ws_b, whereas WD – of states wd_1, wd_2, …, wd_c. CPT of the node WS is a b-table (a table with b entries) containing the probability distribution $\mathrm{P(WS = ws}_e)$, $e = 1,2,\ldots,b$, with $\sum\limits_e \mathrm{P(WS = ws}_e) = 1$. CPT of the node WD is a c-table containing the probability distribution $\mathrm{P(WD = wd}_f)$, $f = 1,2,\ldots,c$, with $\sum\limits_f \mathrm{P(WD = wd}_f) = 1$.



WS	P(WS)
ws_1	P(WS=ws_1)
ws_2	P(WS=ws_2)
⋮	⋮
ws_b	P(WS=ws_b)

WD	P(WD)
wd_1	P(WD=wd_1)
wd_2	P(WD=wd_2)
⋮	⋮
wd_c	P(WD=wd_c)

WS		ws_1			.....	ws_b		
WD		wd_1	.....	wd_c	.....	wd_1	.....	wd_c
WP	wp_1	P(WP=$wp_{1,1,1}$)	.....	P(WP=$wp_{1,1,c}$)	.....	.....	.....	.....
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	wp_s	P(WP=$wp_{s,1,1}$)	.....	P(WP=$wp_{s,1,c}$)	....	.....	.....	.....

Fig. 3. Bayesian network for wind power prediction and the conditional probability tables of the nodes: WS, WD and WP

CPT of the node WP is a table containing all the probability assignments $\mathrm{P}(\mathrm{WP} = \mathrm{wp}_{u,e,f}|\mathrm{WS} = \mathrm{ws}_e, \mathrm{WD} = \mathrm{wd}_f)$, $u = 1,\ldots,s$, and $\sum_u \mathrm{P}(\mathrm{WP} = \mathrm{wp}_{u,e,f}) = 1$.

Learning CPTs amounts essentially to counting data records for different conditions encoded in the network. It means that prior probability distribution of WS, WD and WP can be obtained from relative counts of various outcomes in those data records that meet the conditions described by a combination of the outcomes of the parent variables. The data records in question (wind speed and direction measured by anemometer installed at the top of the turbine nacelle and output power) can be gathered from real data acquired by SCADA system of wind turbine. The resolution of data can be different, but 1 h is very common (then data are average values along 1 h).

BN has a built-in computational architecture for computing the effects of evidence on the states of variables in the model. This architecture allows for updating probabilities of the states of the random variables, on learning new evidence. It means that one can calculate conditional probability function of WP given forecasted wind speed and duration, within time $t = 1, 2, ..,m$. Then $p_{1,i} = p_{1,e,f}$ (for $t = i$) ….. $p_{s,i} = p_{s,e,f}$ (for $t = i$). Conditional probability function of WP needs to be calculated within the time ($t > 0$ –see Fig. 1), i.e. $t = i$ (starting point of the maintenance activity) and $t = i - 1$, $t = i - 2$, … (mandatory for calculating the probability distribution of wind power within $t = i - 1$ and duration of occupying the states within $t = i - 1$). It is assumed that the states occupied within $t = i - 2$, $t = i - 3$, …. are the states of the highest probability $p_{u,i-2}, p_{u,i-3}, \ldots$.

The error of the WPP to be obtained based on BN model depends on the errors of forecasting wind speed and direction (meteorological models) and the number of WPP states in the BN model, the number of data used for learning parameters of BN and marginal distributions of wind speed and direction data.

3.2 The Second Order Semi-markov Chains Used for Forecasting of Wind Turbine Output Power Trajectories

We use a second order semi-Markov chain in state and duration as proposed by [5–7] where additional results can be searched out. The model is used for forecasting wind turbine output power trajectories given the real data of output powers at $t = 0$ and $t = -1$, and duration of output power value at $t = -1$. The time $t = 0$ involves the last unit of time when the output power is acquired whereas $t = -1$ refers to the previous unit. The output power at $t = 0$ is average power within one unit of time up to $t = 0$.

Let us consider a finite set of states $E = \{w_1, w_2, ..., w_S\}$ that represent the different wind turbine output power values; we will often denote the generic wind power value $\boldsymbol{w_j} \in \boldsymbol{E}$ more simply with the symbol j. It should be noted that in this paper we model directly the power output without considering a wind speed model as in [5–7].

Let us also consider a complete probability space (Ω, F, P) on which we define the following random variables:

$$J_n : \Omega \to E, \quad T_n : \Omega \to \mathbb{N}.$$

The variable J_n denotes the wind output power at the n-th transition and T_n is the time of the n-th transition of the power. We assume that

$$\begin{aligned} &P[J_{n+1} = j, T_{n+1} - T_n = t | \sigma(J_s, T_s, 0 \leq s \leq n), J_n = k, J_{n-1} = i, T_n - T_{n-1} = x] \\ &= P[J_{n+1} = j, T_{n+1} - T_n = t | J_n = k, J_{n-1} = i, T_n - T_{n-1} = x] =: {}_x q_{i.k,j}(t). \end{aligned} \tag{6}$$

Relation (6) makes the fact clear that the knowledge of the values J_n, J_{n-1}, $T_n - T_{n-1}$ (the most two recent values of the output power and the length of time between them) suffices to give the conditional distribution of the couple J_{n+1}, $T_{n+1} - T_n$ (next power and length of time necessary to enter in next power) whatever the values of the past variables might be. Therefore, to make probabilistic forecasting we need the knowledge of the last two visited states and the duration time of the transition between them.

Denote by $N(t) = \sup\{n \in \mathrm{IN} : T_n \leq t\}$ $\forall t \in \mathrm{IN}$ the number of transitions held by the output power process up to time t. We define the second order (in state and duration) semi-Markov chain as $\mathbf{Z}(t) = (Z_1(t), Z_2(t)) : = (J_{N(t)-1}, J_{N(t)})$.

For this model ordinary transition probability functions and transition probabilities with initial and final backward recurrence times were defined and computed in [5], reliability measures applied to wind energy production were presented in [6] and equations for higher order moments of the second order semi-Markov reward chain were computed and used to quantify the total energy produced by a wind turbine in a given time interval, see [7].

The probabilistic behaviour of the output power process can be summarized by the transition probability function, namely the function defined by

$${}_x\varphi_{i.k,h,j}(t) := P\left[J_{N(t)} = j, J_{N(t)-1} = h | J_{N(0)} = k, J_{N(0)-1} = i, T_{N(0)} = 0, T_{N(0)} - T_{N(0)-1} = x\right].$$

These probabilities satisfy the following system of equations, see [5]

$$ {}_x\varphi_{i.k,h,j}(t) = 1_{\{i=h,k=j\}}\left(1 - \sum_{j\in E}\sum_{s=1}^{t} {}_x q_{i.k,j}(s)\right) + \sum_{r\in E}\sum_{s=1}^{t} {}_x q_{i.k,r}(s)\cdot {}_s\varphi_{k.r,h,j}(t-s). $$

For our purposes it is sufficient to consider the probability

$$ {}_x\tilde{\varphi}_{i.k,j}(t) := \sum\nolimits_{h\in E} {}_x\varphi_{i.k,h,j}(t). \tag{7} $$

Given the probability function (7) we can compute the forecast of the output power process in the following way. First we note that the forecast of the wind power at a given time $t > 0$ (t steps ahead) depends on the available information at the current time $s = 0$. Therefore, if at time $s = 0$ we know that $J_{N(0)} = k, J_{N(0)-1} = i, T_{N(0)} = 0, T_{N(0)} - T_{N(0)-1} = x$ then the forecast value at time t can be denoted by the ${}_x\hat{Z}_{i.k}(t)$ where

$$ {}_x\hat{Z}_{i.k}(t) = mode\left[Z_t|J_{N(0)} = k, J_{N(0)-1} = i, T_{N(0)} = 0, T_{N(0)} - T_{N(0)-1} = x\right], $$

That is the mode of the conditional distribution of the state occupancy. It is simple now to realize that

$$ {}_x\hat{Z}_{i.k}(t) = \max\nolimits_{j\in E}\left\{{}_x\tilde{\varphi}_{i.k,j}(t)\right\}. \tag{8} $$

The quantity computed in formula (8) is a punctual forecast of the output power. Very often it is important to provide also the so called predictive intervals. To this end we define the random variable called conditional forecasting error

$$ {}_xe_{i.k}(t) := {}_x\hat{Z}_{i.k}(t) - Z(t). $$

In our framework ${}_xe_{i.k}(t)$ is a discrete random variable that assumes values

$$ {}_x\hat{Z}_{i.k}(t) - w_r, \quad \forall w_r \in E, $$

with corresponding probability ${}_x\tilde{\varphi}_{i.k,r}(t)$. This means that if we fix a predictive level α we can compute the predictive interval to the level α by identifying the $\frac{\alpha}{2}$– quantile named pv_1 and the $\left(1-\frac{\alpha}{2}\right)$– quantile denoted by pv_2. Thus the interval

$$ \left({}_x\hat{Z}_{i.k}(t) - pv_1, {}_x\hat{Z}_{i.k}(t) - pv_2\right), $$

gives the predictive interval to the level α.

4 Application to Real Data

4.1 Description of the Case Study and Assumptions

The wind turbine in question is E-48 of rated output power 800 kW by Enercon. Number of data records is 46,000 (duration of operation 5 years and 4 months). Data resolution is 1 h. The data are used for: (i) learning the parameters of BN model and (ii) formulating the second order semi-Markov chain.

The structure of the BN model is already defined (see Fig. 3). To learn the parameters of this network one needs the dataset mentioned above. Each variable is discretized following two assumptions, i.e. a method of discretization (uniform widths and uniform counts) and the number of bins (5 and 10). Having learnt parameters of the network given the method of discretization and the number of bins the validation of the results has been performed. The K-crossvalidation has been taken for it given fold count K = 10. As a result of validation the accuracy, confusion matrix, ROC Curve and Calibration have been obtained. Finally the states of variables WS, WD and WF, and their boundaries are derived (see Table 1) and the model can be applied for reasoning.

Table 1. States of the WS, WD and WP, and boundaries of partitions represented by the states

State	WS		WD		WP	
	From	To	From	To	From	To
S1	0.0	3.2	0.0	84.0	0.0	21.0
S2	3.2	4.1	84.0	138.0	21.0	70.0
S3	4.1	4.7	138.0	163.0	70.0	144.0
S4	4.7	5.2	163.0	185.0	144.0	289.0
S5	5.2	5.7	185.0	208.0	289.0	835.0
S6	5.7	6.3	208.0	231.0		
S7	6.3	7.0	231.0	256.0		
S8	7.0	7.7	256.0	281.0		
S9	7.7	8.8	281.0	313.0		
S10	8.8	18.0	313.0	359.9		

The wind speed and direction forecast and wind turbine output power probability distributions given *t* are provided in Table 2.

The BN model used for derivation of wind turbine output power probability distributions achieved 88.852% accuracy in inferring the correct wind turbine output power (it guessed correctly 88852 out of the total of 100000 records). Concerning individual states the accuracy is as follows: S1 – 87.836%, S2 – 82.070%, S3 – 91.661%, S4 – 86.036% and S5 – 96,754%.

Let us consider two starting points of maintenance $t = 1$ and $t = 2$, and four maintenance durations 5 h, 6 h, 7 h and 8 h. Durations longer than 8 h are uncommon since a wind turbine (or farm) operator decides that the planned maintenance should be completed within one work shift during the day.

Table 2. The WS and WD forecast and WP distributions given *t*

Time		$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$
Forecast	Wind speed	6.5	5.6	4.9	4.4	4.1	3.8
	Wind direction	199	191	181	174	167	160
State	S1	0.0107	0.0017	0.0159	0.0609	0.0609	0.7475
	S2	0.0047	0.1275	0.8325	0.9376	0.9376	0.2519
	S3	0.0420	0.8693	0.1501	0.0014	0.0014	0.0006
	S4	0.9421	0.0000	0.0015	0.0001	0.0001	0.0000
	S5	0.0005	0.0015	0.0000	0.0000	0.0000	0.0000

4.2 Validation of the Wind Turbine Output Power Trajectory Modeling

To validate the model we performed Monte Carlo simulation to generate a synthetic trajectory of output power. For real and synthetic trajectory the sample autocorrelation function was computed for lag up to 200 h. The two sample autocorrelations are almost identical, as shown in Fig. 4, and their mean percentage error equals 2.95%.

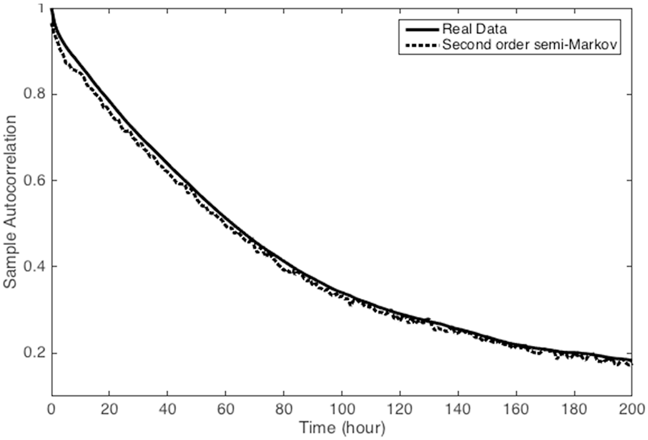


Fig. 4. Sample autocorrelation function for real and synthetic output power trajectories

4.3 Discussion of the Maintenance Scheduling

The 'Expected utility' results of the influence diagram (see Fig. 1) for decision problem of wind turbine maintenance scheduling are depicted in Fig. 4.

According to Fig. 5 the best decision that minimizes the energy not supplied because of disabled wind turbine is to start the maintenance within $t = 2$ rather than $t = 1$. It is valid for each maintenance duration considered in the study. To achieve competitiveness of the maintenance to be begun within $t = 1$ as compared to the other one, the limitation of maintenance duration is needed (at least 1 h). Sometimes it cannot be possible because of restrictions in accessibility of technical and human

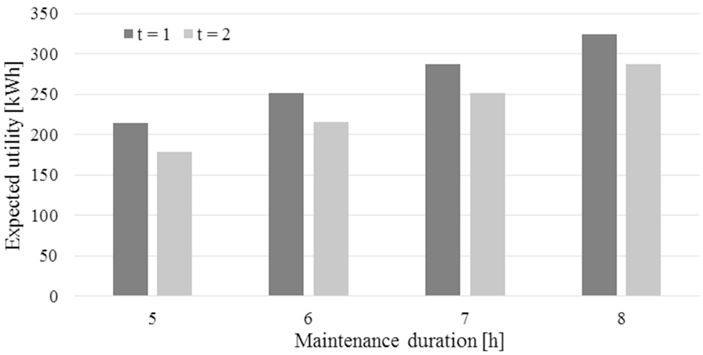


Fig. 5. The 'Expected utility' results of the influence diagram for decision problem of wind turbine maintenance scheduling

resources. In both cases of starting points of maintenance activity the shorter (longer) maintenance duration is the lower (higher) 'Expected utility' one can expect.

5 Conclusion

Wind turbine maintenance is one of the most important activities in the wind energy industry. In this paper we presented methodology that considers maintenance scheduling of wind turbines as a decision making problem. The optimal decision is established in order to minimize the energy not supplied during the maintenance activity. The methodology makes use of an influence diagram where the conditional probability distribution of a chance node of the diagram is obtained relying on Bayesian networks, whereas the utilities of value node are calculated thanks to the second order semi-Markov chains.

The model has important practical implications because it may help wind energy engineers in the planning and execution of maintenance activity following optimal decision rules.

The work has been prepared under the project S/WE/4/13 and financially supported by Ministry of Science and Higher Education, Poland.

References

1. Pattison, D., Segovia Garcia, M., Xie, W., Revie, M., Whitfield, R.I., Irvine, I.: Intelligent integrated maintenance for wind power generation. Wind Energ. **19**, 547–562 (2016)
2. Kerres, B., Fischer, K., Madlener, R.: Economic evaluation of maintenance strategies for wind turbines: a stochastic analysis. IET Renew. Power Gener. **9**(7), 766–774 (2015)
3. Sperstad, I.B., McAuliffe, F.D., Kolstad, M., Sjomark, S.: Investigating key decision problems to optimize the operation and maintenance strategy of offshore wind farms. Energ. Procedia **94**, 261–268 (2016)

4. Kjaerulff, U.B., Madsen, A.L.: Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis. Springer, New York (2008)
5. D'Amico, G., Petroni, F., Prattico, F.: First and second order semi-Markov chains for wind speedmodeling. Phys. A **392**(5), 1194–1201 (2013)
6. D'Amico, G., Petroni, F., Prattico, F.: Reliability measures of second order semi-Markov chain with application to wind energy production. J. Renew. Energ. **2013**, 6 (2013). (368940)
7. D'Amico, G., Petroni, F., Prattico, F.: Performance analysis of second order semi-markov chains: an application to wind energy production. Methodol. Comput. Appl. Probab. **17**, 781–794 (2015)

Deadlock Detection in Distributed Systems Using the IMDS Formalism and Petri Nets

Wiktor B. Daszczuk[1(✉)] and Wlodek M. Zuberek[2]

[1] Institute of Computer Science, Warsaw University of Technology,
Nowowiejska Street 15/19, 00-665 Warsaw, Poland
wbd@ii.pw.edu.pl
[2] Department of Computer Science,
Memorial University, St. John's AlB 3X5, Canada
wlodek@mun.ca

Abstract. Integrated Model of Distributed Systems (IMDS) is a formalism which expresses duality of message passing and resource sharing and which highlights locality, autonomy of distributed elements as well as asynchrony of actions and communication. Combined with model checking, IMDS allows to verify numerous properties of modeled systems. It also provides insights into the behavior of model components (servers and agents) in the form of server view and agent view of the system. IMDS is used in the Dedan verification environment which can detect several types of deadlocks, including communication deadlocks (in the server view) and resource deadlocks (in the agent view). The paper also outlines a mapping of IMDS models into behaviorally equivalent Petri nets, opening the way for many analysis techniques developed for Petri nets to be used for analysis of IMDS models. In particular, structural (siphon-based) methods for deadlock analysis in Petri nets can be used for deadlock detection in IMDS models.

Keywords: Distributed systems · Distributed system modeling · Deadlock detection · Petri nets · Petri net siphons · Formal methods

1 Introduction

IMDS (Integrated Model of Distributed Systems, [1–3]) is a formalism for describing the behavior of distributed systems, especially for finding deadlocks. In IMDS, a communication dualism is exploited, since the modeled system is represented as server processes that communicate by messages, or travelling processes (agents) that communicate by means of servers' states. A model of a distributed system is uniform, it can be decomposed ("cut") to server processes or agent processes. A system consists of actions, which are combined in sequences to form the processes. An action has a server's state and an agent's pending message on input, and which produces similar pair (a new server's state and a new agent's message) on output.

The two views of a system (server view and agent view) are obtained by the two possible groupings of a set of actions into sequences. In the server view, actions in individual servers are grouped into processes. The server's states are the carrier of the

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_12

server's process, and the messages are the communication means between server processes. In the agent view, actions concerning an individual agent conform a process. Messages are internal to a process: they are the carrier of the process. The agent processes communicate via servers' states.

The IMDS formalism was used, together with model checking technique [4], to develop the Dedan program which finds various kinds of deadlock in a verified system [5]. These are: communication deadlock (in the server view), resource deadlock (in the agent view), partial deadlock (in which a subset of system's processes participate) and total deadlock (concerning all processes). A counterexample is generated if a deadlock is found.

In Dedan, automatic conversion between the server view and the agent view is performed. Also, observation of global transition graph and simulation on this graph are possible.

Dedan is built in such a way that the specification of temporal formulas and temporal verification are hidden to a user. The reason is that model checking techniques are seldom known by the engineers. Therefore, the program is constructed in such a way that a user specifies the system and simply "pushes the button" to check for the existence of deadlocks.

The model checking technique has a disadvantage: the evaluation of temporal formula consists in finding a single global configuration (will be defined in Sect. 2) providing a false result is enough. The designer may repair the erroneous specification and run the verification again. The scheme may be repeated for many times, until all deadlocks are found and repaired.

The other technique of deadlock identification is finding siphons in a Petri net corresponding to a verified IMDS specification. A siphon is a Petri subnet, which cannot restore tokens if it is emptied [6–8]. If an empty siphon is reachable, it denotes a deadlock. The deadlock concerns the processes (server processes and/or agent processes) that take part in the siphon. Therefore, it may be total or partial deadlock.

A conversion of IMDS to a Petri net is described in Sect. 4 [1]. This allows for finding some structural properties of a verified system, like identification of dead code (unreachable part of a net), invariants discovery, finding separated subnets (independent subsystems that do not cooperate) or location of siphons which help to find all possible deadlocks in a system, not only one as in model checking techniques.

The main contribution of the paper is the integration of the IMDS formalism, preserving communication duality, locality and autonomy of distributed elements, and asynchrony of actions and communication, with siphon identification technology, allowing for finding all deadlocks in verified system.

In this paper a definition of IMDS is given in Sect. 2, and the example of a bounded buffer is in Sect. 3. The conversion of IMDS specification to a Petri net is covered in Sect. 4. An example of deadlock detection using siphons and reachability is described in Sect. 5. A problem with siphons in a system with leader is covered in Sect. 6. Section 7 concludes the research.

2 Integrated Model of Distributed Systems (IMDS)

IMDS is defined in [1–3]. Here we use simplified version of IMDS, without dynamic process creation, which is suitable for static model checking. A system is composed of a finite set of *servers* with their *states* as pairs *(server, value)*. Any server is equipped with a finite set of *services*. Servers communicate by *messages (agent, server, service)*, where *agents* are distinguishable distributed computations. States and messages are together called *items*. A set of states and messages, one state for any server and at most one message for any agent is a *global system configuration* (or *configuration* in short). This configuration is concerned as a global system state (but we reserve a term *state* for servers only): set of *current* servers' states and *pending* agents' messages, except for terminated agents. The *initial configuration* consists of *initial states* of all servers and *initial messages* of all agents. Formally:

$S = \left\{s_1, s_2, \ldots, s_{card(S)}\right\}$ – finite set of servers
$A = \left\{a_1, a_2, \ldots, a_{card(A)}\right\}$ – finite set of agents
$V = \left\{v_1, v_2, \ldots, v_{card(V)}\right\}$ – finite set of values
$R = \left\{r_1, r_2, \ldots, r_{card(R)}\right\}$ – finite set of services
$P \subset S \times V$ – set of states
$M \subset A \times S \times R$ – set of messages
$I = P \cup M$ – set of items

$T \subset I; \forall_{p,p' \in T \cap P} \, p = (s,v),\, p' = (s',v'),\ p \neq p' \Rightarrow s \neq s';$
$\forall_{m,m' \in T \cap M} \, m = (a,s,r),\, m' = (a',s',r'),\ m \neq m' \Rightarrow a \neq a'\}$ – configuration; one state for every server, at most one message for every agent

$T_0 \subset I; \forall_{s \in S} \exists_{p \in T_0 \cap P} \, p = (s,v),\, v \in V;$
$\forall_{a \in A} \exists_{m \in T_0 \cap M} \, m = (a,s,r),\, s \in S,\, r \in R\}$ – initial configuration; one state for every server, one message for every agent

A set of *actions* is defined by a relation Λ. An action is defined for a pair *(message, state)*, it retrieves the *input state* and the *input message* from the *input configuration* and inserts a new state and a new message (or a new state only in the case of agent-terminating action) to an *output configuration*. We say that a pair *(message, state) match* if an action is defined for the pair. In a matching pair *(message, state)*, the server component in message and state must be the same – this means that the service is invoked on the server the message is pending on. It is obvious that the server component of the input state and the *output state* is the same, which means that an output state refers to the same server as the input one. An *output message* (if any) is in the context of the same agent as the input one, but the output message is typically directed to some other server (it may be directed to the same server as well). Formally:

$\Lambda = \{(m,p)\, \lambda(m',p')\} \cup \{(m,p)\, \lambda(p')\} \mid$
$m = (a,s,r) \in M,\, m' = (a',s'',r') \in M,\ a' = a;$
$p = (s,v) \in P,\, p' = (s',v') \in P,\ s' = s$ – set of actions – ordinary and agent-terminating

$$\begin{aligned} T_{inp}(\lambda) &= T \mid T \supset \{m,p \mid (m,p)\,\lambda(m',p') \vee (m,p)\,\lambda(p')\} - \text{input configuration of action } \lambda; \\ &\quad \text{contains input items of the action} \end{aligned}$$

$$\begin{aligned} T_{out}(\lambda) &= T \mid T \supset \{m',p' \mid (m,p)\,\lambda(m',p')\} \vee \\ &\quad T \supset \{p' \mid (m,p)\,\lambda(p')\} \wedge (m = (a,s,r),\ \forall_{m'=(a',s',r') \in M}\, a' = a \Rightarrow m' \notin T) - \text{output} \\ &\quad \text{configuration of action } \lambda; \text{ contains output items of the action, does not contain} \\ &\quad \text{output message of the agent in a case of agent-terminating action} \end{aligned}$$

A *Labeled Transition System* (LTS [9]) represents the behavior of a system as it contains all executions of the system. *Nodes* (not called *states* for unambiguousness) are global system configurations and *transitions* are actions. Formally:

$$\begin{aligned} \delta = \{ & \delta(\lambda) \mid \lambda \in \Lambda,\ T_{inp}(\lambda)\,\delta(\lambda)\,T_{out}(\lambda), \\ & (((a,s,r),\,(s,v))\,\lambda((a,s',r'),\,(s,v')) \vee ((a,s,r),\,(s,v))\,\lambda((s,v'))) \wedge \\ & \forall_{a'' \in A}\, a'' \neq a \wedge (a'',s'',r'') \in T_{inp}(\lambda) \Rightarrow (a'',s'',r'') \in T_{out}(\lambda) \wedge \\ & \forall_{s''' \in S}\, s''' \neq s \wedge (s''',v''') \in T_{inp}(\lambda) \Rightarrow (s''',v''') \in T_{out}(\lambda)\,\} - \text{ succession relation;} \\ & \text{replaces input items by output items, all other items preserved} \end{aligned}$$

$$\begin{aligned} LTS = &<N, N_0, W> \mid \\ & N = \{T_0, T_1, \ldots\}(\text{nodes}); \\ & N_0 = T_0(\text{initial node}); \\ & W = \{\delta(\lambda) \mid \lambda \in \Lambda,\ T_{inp}(\lambda)\ \delta(\lambda)\ T_{out}(\lambda) \wedge \\ & (T_{inp}(\lambda) = T_0 \vee \exists_{\lambda' \in \Lambda}\, \delta(\lambda') \in W,\ T_{out}(\lambda') = T_{inp}(\lambda))(\text{transitions})\} - \text{labeled transition system} \end{aligned}$$

Actions are executed in interleaving way (one action at a time [10]). Note that every server performs its action autonomously (only the server's state and the messages pending on this server are considered). Also, the communication is asynchronous: a server process sends a message to some other server process (or an agent sets the server's state for some other agent) regardless of the current situation of a process with which it communicates (and every other process). As a result, we may call the process *autonomous* and *asynchronous*.

The processes in the system are defined as *sequences of actions*. If two consecutive actions in a process are connected by a server state – it is a *server process* communicating with other server processes by means of messages (states are the carriers of server processes). If two consecutive actions in a process are connected by a message of an agent – it is an *agent process* communicating with other agent processes by means of servers' states (agent messages are the carriers of agent processes). Server processes or agent processes are extracted by simply grouping of actions: a server process is a set of actions having its states on input (the output state concerns the same server as the input state). An agent process is a set of actions having its messages on input (if there is an output message, it concerns the same agent).

The decomposition of a system into server processes is called a *server view*, the other one is an *agent view*. Formally:

$B(s) = <\lambda_1, \lambda_2, \cdots > |$
$((a,s,r),(s,v))\,\lambda_1((a,s',r'),\,(s,v'))\,\vee\,((a,s,r),(s,v))\,\lambda_1((s,v')),$
$((a',s',r''),\,(s,v'))\,\lambda_2((a',s'',r'''),(s,v''))\,\vee\,((a',s',r''),(s,v'))\,\lambda_2((s,v''))$ – server process of server s

$C(a) = <\lambda_1, \lambda_2, \cdots > |$
$((a,s,r),(s,v))\,\lambda_1((a,s',r'),(s,v')),((a,s',r'),(s',v''))\lambda_2((a,s'',r''),(s',v'''))\,\vee$
$<\lambda_1, \lambda_2, \ldots, \lambda_k > |$
$((a,s,r),(s,v))\lambda_1((a,s',r'),(s,v')),((a,s',r'),(s',v''))\lambda_2((a,s'',r''),(s',v''')),$
$((a,s'''',r''''),(s'''',v''''))\lambda_k((s'''',v'''''))$ – agent process of agent a(non-terminating or terminating)

$\boldsymbol{B} = \{B(s)\}\,|\forall_{s \in S} B(s) \in \boldsymbol{B}$ – server view; decomposition of a system into server processes

$\boldsymbol{C} = \{C(a)\}\,|\forall_{a \in A} C(a) \in \boldsymbol{C}$ – agent view; decomposition of a system into agent processes

The deadlocks in a system are defined as follows:

- a *communication deadlock* of a server process – when there are messages pending at the server, but no matching pair of any message with the server state will occur;
- a *resource deadlock* of an agent process – when an agent's message is pending at a server but it will never match any current or future state of this server.

The universal temporal formulas find the deadlocks [2, 3]: communication deadlocks in the server view and resource deadlocks in the agent view. Therefore, temporal logic is hidden in the verification tool and the user need not specify temporal formulas to find deadlocks. In [3], an example is described of a system of two semaphores used by two agents (and third server doing something else to show that a deadlock may be partial). A detection of deadlocks in the server view (from the perspective of semaphores) and in the agent view (from the perspective of semaphore users) are presented. The other example is Automatic Vehicle Guidance System [11]. The server view shows the cooperation of road segment controllers while guiding a vehicle, and the agent view shows the movement from the perspective of the vehicles.

3 Simple Example – Buffer

To present the two views in IMDS source language, a simple system containing a buffer with producer and consumer agents (each one originating from its own server) is included below. The system is listed in the server view. The notation is intuitional: server types are defined (lines 3, 12, formal parameters specify agents and other servers used). Every server includes states (l.4, 13), services (l.5, 14) and actions (l.7–10, 16–19) (an action *((a, s, r),(s, v))λ((a, s ', r '),(s,v '))* has the form `{a.s.r, s.v}→{a.s′.r′, s.v′}`). Then, server and agent variables are declared (l.21–22). The variables have the same names as the types, they are distinguished by context. If a variable has the same identifier as its type, a declaration `variable:type` may be suppressed to a single identifier, as in the example. At the end, servers (l.24–25) and agents (l.26) are initialized (and variable names are bound with formal parameters of servers).

```
1.   #DEFINE N 2
2.   #DEFINE K 1

3.   server:  buf(agents Aprodcons[N];servers Sprodcons[N]),
4.   services {put, get},
5.   states   {elem0,elem[K]},

6.   actions  {
7.   <i=1..N>{Aprodcons[i].buf.put, buf.elem0}              ->
              {Aprodcons[i].Sprodcons[i].ok_put, buf.elem[1]},
8.   <i=1..N><j=1..K-1>{Aprodcons[i].buf.put, buf.elem[j]}  ->
              {Aprodcons[i].Sprodcons[i].ok_put, buf.elem[j+1]},
9.   <i=1..N><j=2..K>{Aprodcons[i].buf.get, buf.elem[j]}    ->
              {Aprodcons[i].Sprodcons[i].ok_get, buf.elem[j-1]},
10.  <i=1..N>{Aprodcons[i].buf.get, buf.elem[1]}            ->
              {Aprodcons[i].Sprodcons[i].ok_get, buf.elem0}
11.  };

12.  server:  Sprodcons(agents Aprodcons;servers buf),
13.  services {doSth,ok_put,ok_get}
14.  states   {neutral,prod,cons}
15.  actions  {
16.  {Aprodcons.Sprodcons.doSth, Sprodcons.neutral} ->
              {Aprodcons.buf.put, Sprodcons.prod}
17.  {Aprodcons.Sprodcons.doSth, Sprodcons.neutral} ->
              {Aprodcons.buf.get, Sprodcons.cons}
18.  {Aprodcons.Sprodcons.ok_put, Sprodcons.prod}    ->
              {Aprodcons.Sprodcons.doSth, Sprodcons.neutral}
19.  {Aprodcons.Sprodcons.ok_get, Sprodcons.cons}    ->
              {Aprodcons.Sprodcons.doSth, Sprodcons.neutral}
20.  };

21.  servers  buf,Sprodcons[N];
22.  agents   Aprodcons[N];

23.  init     ->       {
24.  <j=1..N> Sprodcons[j](Aprodcons[j],buf).neutral,
25.           buf(Aprodcons[1..N],Sprodcons[1..N]).elem0,
26.  <j=1..N> Aprodcons[j].Sprodcons[j].doSth,
27.  }.
```

There are two obvious deadlocks in the example: when the two agents both try to `get` from an empty buffer and when they try to `put` to a full buffer. The former one is shown in the counterexample presented in Fig. 1. However, it is hard to “enforce” the model checker to show the latter deadlock (if it is even known by the designer).

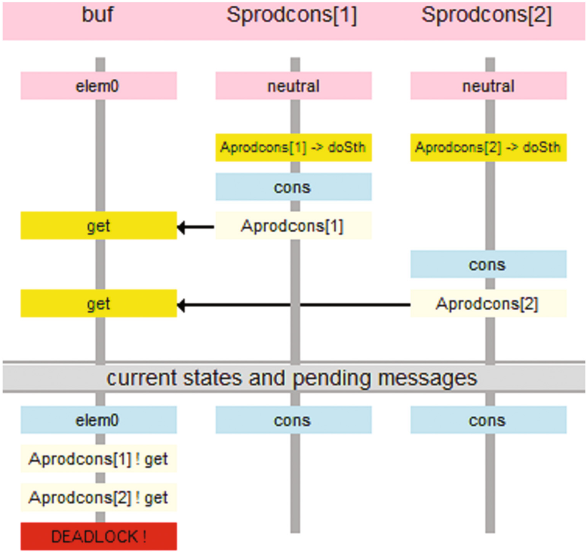


Fig. 1. The counterexample of a deadlock in “bounded buffer” system

4 Petri Net Equivalent to IMDS

Although the main target of Dedan is finding deadlocks, a user may be interested in other properties of a verified system, for example:

- structural properties of a system: structural conflicts, dead code, pure cyclic system or not, etc.,
- temporal properties other than deadlock: if a system is safe from some erroneous situation, if given situations are inevitable, etc.,
- graphical definition of concurrent components of a system (servers or agents),
- graphical simulation in terms of concurrent components rather than in terms of a global graph.

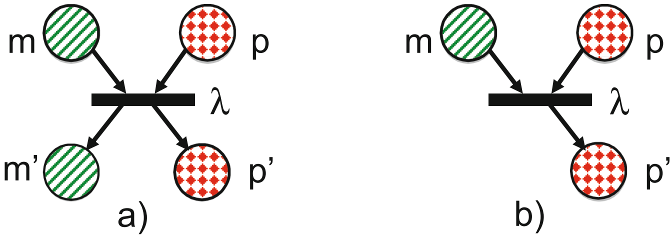


Fig. 2. Petri net interpretation of (a) regular action (b) agent-terminating action

For the purpose of supporting the above possibilities, some additional facilities are included into Dedan. Besides the export to external model checkers, for temporal analysis, the interface with Charlie Petri net analyzer [12, 13] is included, to obtain a structural analysis. The export is in ANDL format (*Abstract Net Description Language* [14]).

An IMDS system may be converted to an equivalent Petri net. Every action is converted to a Petri net transition, as illustrated in Fig. 2. Input items (a message *m* and a state *p*) are converted to the input places *m* and *p*. In a regular action, output items (a message m' and a state p') are converted to the output places m′ and p′ (Fig. 2a). In an agent-terminating action, only one output place is present (corresponding to an output state p′, Fig. 2b). The initial marking of the Petri net has tokens in all places of initial servers' states and all places of initial agents' messages. By construction of the described conversion of an IMDS system to a Petri net, the reachable markings graph has identical structure as LTS of IMDS (states↔"state" places, messages↔"message" places, actions↔transitions, configuration↔marking, initial configuration↔initial marking).

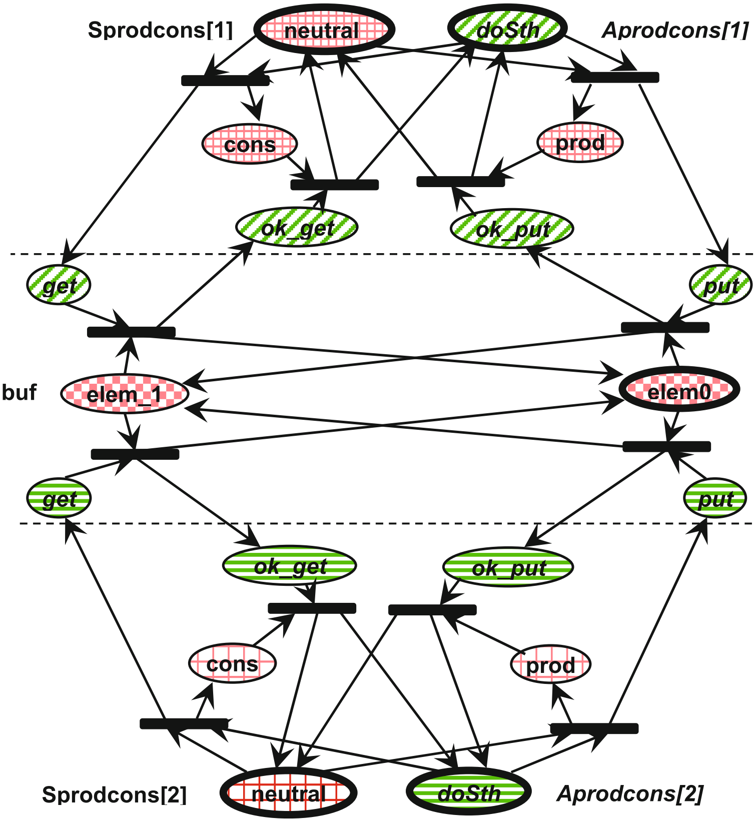


Fig. 3. Petri net representation of the "bounded buffer" system: servers *Sprodcons[1..2]*, *buf*, agents *Aprodcons[1..2]*

The “bounded buffer” system converted to the Petri net (in ANDL format) is shown below. First, the places with initial marking are defined. Then the transitions follow, with input places marked “−1” and output places “+1”. The final “1” is not important. The indices of vector elements (in square brackets) are replaced by indices separated from vector names with underscore. Identifiers in messages *(agent, server, service)* and states *(server, value)* are separated also by underscores. For example, a message *(Aprodcons[1], buf, put)* is converted to place identifier `Aprodcons_1_buf_put`.

```
pn [ IMDS ] {
constants:
places:
Aprodcons_1_buf_put = 0 ;
Aprodcons_1_buf_get = 0 ;
Aprodcons_2_buf_put = 0 ;
Aprodcons_2_buf_get = 0 ;
buf_elem0 = 1 ;
buf_elem_1 = 0 ;
…
transitions:
buf_1 : : [Aprodcons_1_buf_put - 1] & [buf_elem0 - 1] &
          [Aprodcons_1_Sprodcons_1_ok_put + 1] & [buf_elem_1 + 1] : 1 ;
buf_2 : : [Aprodcons_2_buf_put - 1] & [buf_elem0 - 1] &
          [Aprodcons_2_Sprodcons_2_ok_put + 1] & [buf_elem_1 + 1] : 1 ;
buf_3 : : [Aprodcons_1_buf_get - 1] & [buf_elem_1 - 1] &
          [Aprodcons_1_Sprodcons_1_ok_get + 1] & [buf_elem0 + 1] : 1 ;
buf_4 : : [Aprodcons_2_buf_get - 1] & [buf_elem_1 - 1] &
          [Aprodcons_2_Sprodcons_2_ok_get + 1] & [buf_elem0 + 1] : 1 ;
Sprodcons_1_5 : : [Aprodcons_1_Sprodcons_1_doSth - 1] &
          [Sprodcons_1_neutral - 1] & [Aprodcons_1_buf_put + 1] &
          [Sprodcons_1_prod + 1] : 1 ;
Sprodcons_1_6 : : [Aprodcons_1_Sprodcons_1_doSth - 1] &
          [Sprodcons_1_neutral - 1] & [Aprodcons_1_buf_get + 1] &
          [Sprodcons_1_cons + 1] : 1 ;
…
}
```

The Petri net is illustrated in Fig. 3. The states and messages in individual servers are grouped and separated by dashed lines. The states of servers are filled red while the messages are filled green. Also, *`Sprodcons[1]`* states have dense grill while *`Sprodcons[2]`* states have rare grill. States of *`buf`* have chessboard filling. Messages of *`Aprodcons[1]`* have diagonal hatching while messages of *`Aprodcons [2]`* have horizontal hatching. Initial states and initial messages are surrounded by bold ovals. All messages have identifiers in italics.

5 Deadlock Detection Using Siphons and Reachability

The siphon report from Charlie [13] contains (among other data):

```
minimal siphon ( place ) =
1             |0.Aprodcons_1_buf_put:1,
              |2.Aprodcons_2_buf_put:1,
              |5.buf_elem_1  :1,
              |6.Aprodcons_1_Sprodcons_1_doSth     :1,
              |8.Aprodcons_1_Sprodcons_1_ok_get    :1,
              |10.Sprodcons_1_prod  :1,
              |14.Aprodcons_2_Sprodcons_2_ok_get   :1,
              |15.Sprodcons_2_neutral       :1,
              |16.Sprodcons_2_prod  :1
2             |1.Aprodcons_1_buf_get:1,
              |3.Aprodcons_2_buf_get:1,
              |4.buf_elem0   :1,
              |6.Aprodcons_1_Sprodcons_1_doSth     :1,
              |7.Aprodcons_1_Sprodcons_1_ok_put    :1,
              |11.Sprodcons_1_cons  :1,
              |13.Aprodcons_2_Sprodcons_2_ok_put   :1,
              |14.Aprodcons_2_Sprodcons_2_ok_get   :1,
              |15.Sprodcons
              … (etc)
```

As the system has deadlocks, there should be siphons that may be emptied, representing the two deadlocks. There are 49 elementary siphons found. Every empty siphon may be checked for reachability using model checking technique (in the Charlie's output, a set of places is listed that should be emptied, and every emptied place represents the lack of a message or a state in a configuration), using the CTL formula ***AG*** *(not* φ*)* (or LTL formula □*(not* φ*)*), where φ is an empty siphon. In the example above, the Uppaal formula (Uppaal [15] is one of external model checkers used in Dedan) for the siphon number 1 is (services has the encoding 1-put, 2-get, 1-*doSth*, 2-*ok_put*, 3-*ok_get*):

```
A[] !(Aprodcons_1_buf!=1 & Aprodcons_2_buf!=1 & !buf_.elem_1
& Aprodcons_1_Sprodcons_1!=1 & Aprodcons_1_Sprodcons_1!=3 &
!Sprodcons_1.prod & Aprodcons_2_Sprodcons_2!=3 &
!Sprodcons_2.neutral & !Sprodcons_2.prod)
```

The verification gives the result *false*, which means that the siphon may be emptied. Uppaal generates a counterexample, in which both agents perform get on empty buffer (state *elem0*). Many other siphons may be emptied by two get operations on the empty buffer. All these situations constitute a single deadlock (but the counterexamples may differ in the order of issuing get by the two agents). Therefore, every siphon may be easily checked for emptying (there are some siphons that are not emptied in the example, for instance *(buf_elem0,buf_elem_1)*). As some siphons are equivalent (they have the same or equivalent counterexamples, differing in the order or operations), abstraction classes should be made in the set of siphons. The criterion for allocation of siphons to abstraction classes is a configuration that finishes a counterexample (it is equivalent to a marking in the Petri net). In the example, there are two abstraction classes representing the two possible deadlocks, they are identified by the configurations (triples are messages, pairs are servers' states):

```
1. {(Aprodcons[1],Sprodcons[1],get),
   (Aprodcons[2],Sprodcons[2],get), (Sprodcons[1],cons),
   (Sprodcons[2],cons), (buf,elem0)}
2. {(Aprodcons[1],Sprodcons[1],put),
   (Aprodcons[2],Sprodcons[2],put), (Sprodcons[1],prod),
   (Sprodcons[2],prod), (buf,elem[1])}
```

For small systems, for which an LTS may be prepared internally in Dedan, the reachability of siphon emptying may be checked directly in the LTS. For instance for the siphon number 1 the reachability of siphon emptying may be verified by the inspection of the LTS and finding a configuration in which:

- pending message of the agent *Aprodcons[1]* is not *put* at *buf*, or the agent is terminated (no pending message of *Aprodcons[1]*),
- pending message of the agent *Aprodcons[2]* is not *put* at *buf*,
- current state of the server *buf* is not *elem_1*,
- pending message of the agent *Aprodcons[1]* is not *doSth* at *Sprodcons[1]*,
- pending message of the agent *Aprodcons[1]* is not *ok_get* at *Sprodcons[1]*,
- current state of the server *Sprodcons[1]* is not *prod*,
- pending message of the agent *Aprodcons[2]* is not *ok_get* at *Sprodcons[2]*,
- current state of the server *Sprodcons[2]* is not *neutral*.

6 Verification of a System with Leader

An example in Sect. 5 was a purely cyclic system. If a system is not cyclic, it may contain a *leader* – a subnet initializing the system before it works cyclically. Such a leader obviously contains a siphon – an emptied subnet which do not restore tokens (because it is not needed). Such a system of “2 semaphores” is described in [3]. A siphon is found for every place implementing an initial state of a server or an initial message of an agent:

```
7              |19.A_2_proc_2_start  :1
8              |12.A_1_proc_1_start  :1
9              |28.r_res      :1
               … (etc)
```

For the siphon number 7, the formula `A[ ] !(A_2_proc_2! = 1)` evaluates to *false*, suggesting a deadlock, but it is an initial message of the agent *A[2]*: *(A[2], proc, start)*, contained in the leader. Distinguishing between deadlock siphons and leader siphons is obvious: if a siphon consists in a single place with no input arcs, it is a leader siphon. As the siphon may be emptied, the initial marking is followed by some other marking and therefore it is not a leader siphon and a deadlock siphon at the same time. There may be some other patterns in LTS (such as initial loops, for example), so further research on this issue is needed.

7 Conclusions

The paper overviews the IMDS formalism and illustrates some of its aspects by discussing a small example of bounded buffer used by producer and consumer agents. Also, a conversion of IMDS models into Petri nets is outlined. Due to this conversion, a wide variety of Petri net model analysis methods can be used for verification of IMDS models. In particular, structural analysis of Petri nets can be used which is not readily available in the IMDS formalism.

There are several aspects of the IMDS formalism that require further research.

One is the usefulness of the IMDS approach for the analysis of large models and the restrictions which the approach may impose on the analyzed models.

For large models, it would be attractive to have some sort of compositional approach in which consecutive model components could be added to the analyzed system without the necessity of reevaluating the system for each addition.

An interesting research question is if the structural methods used for analysis of Petri nets can be adopted to the IMDS formalism; i.e., is it possible to develop structural approach directly in the IMDS formalism, without deriving the Petri net model?

References

1. Chrobot, S., Daszczuk, W.B.: Communication dualism in distributed systems with Petri net interpretation. Theor. Appl. Inform. **18**(4), 261–278 (2006)
2. Daszczuk, W.B.: Deadlock and termination detection using IMDS formalism and model checking, Version 2. ICS WUT Research Report No. 2/2008 (2008)
3. Daszczuk, W.B.: Communication and resource deadlock analysis using IMDS formalism and model checking. Comput. J. (2016) (in press). doi:10.1093/comjnl/bxw099
4. Clarke, E.M., Grumberg, O., Peled, D.: Model Checking. MIT Press, Cambridge (1999). ISBN 0-262-03270-8
5. Dedan. http://staff.ii.pw.edu.pl/dedan/files/DedAn.zip
6. Reisig, W.: Petri Nets - An Introduction. Springer, Heidelberg (1985). ISBN 978-3-642-69970-2
7. Craig, D.C., Zuberek, W.M.: Two-stage siphon-based deadlock detection in Petri nets. In: Petratos, P., Dandapami, P. (eds.) Current Advances in Computing, Engineering and Information Technology, pp. 317–330. International Society for Advanced Research, Palermo, Italy (2008). ISBN 978–960-6672-34-7
8. Chu, F., Xie, X.-L.: Deadlock analysis of Petri nets using siphons and mathematical programming. IEEE Trans. Robot. Autom. **13**(6), 793–804 (1997). doi:10.1109/70.650158
9. Reniers, M.A., Willemse, T.A.C.: Folk theorems on the correspondence between state-based and event-based systems. In: Černá I. et al. (ed.) 37th Conference on Current Trends in Theory and Practice of Computer Science, Nový Smokovec, Slovakia, 22–28 January 2011, pp. 494–505. Springer, Heidelberg (2011). doi:10.1007/978-3-642-18381-2_41
10. Penczek, W., Szreter, M., Gerth, R., Kuiper, R.: Improving partial order reductions for universal branching time properties. Fundam. Informaticae **43**(1–6), 245–267 (2000). doi:10.3233/FI-2000-43123413

11. Czejdo, B., Bhattacharya, S., Baszun, M., Daszczuk, W.B.: Improving resilience of autonomous moving platforms by real-time analysis of their cooperation. Autobusy-TEST **17**(6), 1294–1301 (2016)
12. Charlie: Charlie Petri net analyzer. http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Charlie
13. Heiner, M., Schwarick, M., Wegener, J.-T.: Charlie – an extensible Petri net analysis tool. In: 36th International Conference, PETRI NETS 2015, Brussels, Belgium, 21–26 June 2015, pp. 200–211. Springer, Cham (2015). doi: 10.1007/978-3-319-19488-2_10
14. Schwarick, M., Heiner, M., Rohr, C.: MARCIE - model checking and reachability analysis done efficiently. In: 2011 Eighth International Conference on Quantitative Evaluation of Systems, Aachen, Germany, 5–8 September 2011, pp. 91–100. IEEE, Los Angeles, CA (2011). doi:10.1109/QEST.2011.19
15. Behrmann, G., David, A., Larsen, K.G.: A tutorial on Uppaal 4.0. Report, University of Aalborg, Denmark (2006)

Scheduling Tasks in Embedded Systems Based on NoC Architecture Using Simulated Annealing

Dariusz Dorota(✉)

Cracow University of Technology, Krakow, Poland
ddorota@pk.edu.pl

Abstract. This paper presents a new method to generate and schedule tasks in the architecture of embedded systems based on the simulated annealing. This novel method takes into account the attribute of divisibility of tasks. The paper describes methods in the following chapters in order to generate target system with established restrictions (deadlines). The research activities are an extension of the research presented in the article [1]. As in the case of said work studies, in researches the same algorithms was used for each considered case. Previous studies have indicated very promising results after applying the attribute of divisibility for tasks represented in the created systems.

Keywords: Scheduling · NoC architecture · Simulated annealing

1 Introduction

The necessity to increase productivity and continuous miniaturization within embedded systems enforces continuous work in this area [4–6]. It represents the development of embedded systems, speaks for multi-criteria optimization, which is aimed at creating architectures of embedded systems optimized in terms of criteria, such as the cost of setting up the system, power, speed and design time. These criteria must be met with systems characterized by the increasing complexity [6]. Specified parameters must be taken into account in studies concerning the creation of specialized architectures using appropriate computer methods.

As co-design is defined by parallel designs of both hardware and software components, the effect of which is embedded system implementing their tasks by using integrated components. Usually, optimization undergoes several criteria. In some cases, the aim is to obtain a system that meets one of the criteria with the given limits for the other criteria [6, 7]. There are algorithms that minimize execution time of the system at limited cost, the cost of creating a system at a given speed or to minimize power consumption while providing a minimum speed. Similar algorithms process of co-design is used in this work [8].

This novel method takes into account the attribute of divisibility of tasks. A proposal of representing the process is made in the form of trees. Despite the fact that the architecture of Network-on-Chip (NoC) is an interesting alternative to a bus architecture based on multi-processors systems, it requires a lot of work, which ensures the

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_13

optimization of communication. This paper proposes an effective approach to generate dedicated NoC topology solving communication problems. The research activities are an extension of the research presented in the article [1].

In this work graphs are accepted as TGFF input proposed by David Rhodes and Robert Dick [2, 3].

2 Scheduling

Scheduling problems are applied in the areas of computer science research discrete optimization, discrete programming and combinatorial [9]. For each of the areas becomes the goal of finding the optimum, accepted for the objective function to order tasks using a set of machines for which the process is executed. This process should include the restrictions imposed on the tasks, individual machines and their mutual relations. A typical performance of the scheduling process. In the scheduling problem, two sets can be distinguished:

- $T = \{T_1, T_2, \ldots, {}_n\}$ – tasks set that need to be sorted
- $M = \{M_1, M_2, \ldots, {}_n\}$ – set of machines/processors used to perform tasks

Such criteria as task dependency and independence are also being considered. Whenever term “dependency” is used, it refers to occurrence of dependencies in the order in which tasks are carried out. In case of this criterion, it can be concluded that there are restrictions on the order of tasks. Otherwise, tasks are independent. If dependent tasks are under consideration, they can be presented as a task graph, which makes it possible to illustrate specific links between the tasks. Another task scheduling criterion, quite rare in case of embedded systems, which is examined here, is task divisibility. If a task can be interrupted in the course of performing it, we say that it is divisible. Otherwise it is indivisible.

Typically it is considered in the case of operating systems, where it is possible to interrupt the performance of a given task, and his resume after the completion of the tasks of higher priority. The task can be suspended in the event of a higher priority, or at the moment of the interrupt request. In this work only the case of suspension of the task in the event of a higher priority tasks will be considered. Of course, if such an operation was possible, the task must have the attribute of divisibility. Typically, these attributes are not considered when scheduling tasks within embedded systems. In this article, in addition to the above-described case are taken into account time constraints. Time constraints are set for the selected tracks in the graph. It is one of the main objectives of scheduling. Figure 1 shows the limitations imposed on each path in the graph.

Two types of optimization algorithms can be distinguished in scheduling: algorithms using task succession graphs or algorithms using task interaction graphs. Among the former algorithms are, inter alia, letter schedulers, and among the latter, genetic algorithms.

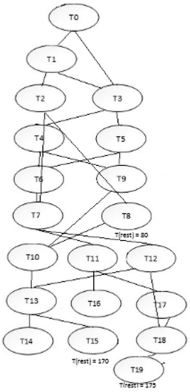


Fig. 1. Elementary task graph

3 Synthesis of SoC Systems

Contemporary embedded systems are characterized by greater complexity, stemming from the integration of constantly increasing diversity of functions in a single system. In addition to the increasing complexity, another aspect of the development of embedded systems are constantly growing requirements related to cost, power consumption, speed of action and the time of design.

3.1 Synthesis and Co-synthesis

The designing of architecture using computer methods specifies the term system synthesis. If such a synthesis is used for embedded systems which integrate components in hardware and software it is called hardware and software co-synthesis [10]. As co-design is defined parallel design of both hardware and software components, the effect of which is embedded system implementing their tasks are using integrated components. Usually optimization undergoes several criteria. In some cases, the aim is to obtain a system that meets one of the criteria with the given limits for the other criteria [7]. There are algorithms that minimize execution time of the system at limited cost, the cost of create a system at a given speed or to minimize power consumption while providing a minimum speed. Similar algorithms process of co-design used in this work [8]. Some work on the co-design of embedded systems [7] allows the use of a simplified architecture which includes a general purpose processor and specialized module.

Methods of co-design based on the specifications presented above form of communicating processes, automatically generate the architecture of the embedded system. This process usually is performed iteratively by making various divisions of functionality and comparing the resulting systems. If an example is considered in which mapping can be used in different processors and hardware modules [7] of co-design process can be divided into the following tasks:

- Defining the architecture through the allocation of modules
- Assignment of tasks to resources
- Scheduling tasks and transmissions

The allocation of computational modules is a step which defines the system architecture. This step is possible after determining the available library modules and their parameters. The next step, usually performed in parallel with the allocation, the allocation of tasks to processors is a generalized problem of the division of tasks between hardware and software. Tasks that are assigned to the processors are subjected to the process of scheduling which is responsible for determining the order in performance of specific tasks. The last two stages, both assigning tasks to processors and scheduling is NP-complete problems. In the class of NP-complete problems finding the target architecture is only possible by using effective heuristic methods.

Many of the proposed co-synthesis methods assumed multi bus architecture. In recent years, single-chip network architecture NoC [4] for systems-oriented intensive calculations was proposed. Many studies concluded that it is most effective architecture for this type of systems.

3.2 NoC Synthesis

Depending on application, and ultimately generated NoC topology, synthetic methods can identify the region in which the graph is synthesized in regular/irregular architectures and each group of the task graph is synthesized in the architecture of the structure of regular/irregular. By selecting regular NoC topology is assumed to simplify the design process. In case of such design decisions it narrows the search for a subset of solutions. Previously, the algorithms took graph tasks as input. This approach allows obtaining the benefits as well as certain restrictions. Approach which used a task graph (TG) as a representation of the data, was used to good effect in [10, 11]. A large number of existing algorithms co-synthesis is a refining [13, 14], such solutions starting from sub-optimal solutions, in the course of operations proposed system is improved by modifying the architecture. In such algorithms solution is usually the initial architecture for the system fulfills the present limit, and then striving to achieve the best system follows the migration process for existing PE, reduction or addition of PE.

Restrictions imposed by heuristic algorithms have been offset by the use in the synthesis of embedded systems metaheuristic algorithms [13]. Simulated annealing algorithm presented in many works multi-purpose research approach based on mapping the mesh structure of NoC architecture. Here the heuristics used in the deterministic algorithm to explore the space and finding the Pareto mappings that optimize performance and power consumption. With the construction of the NoC network, it is possible to virtualize hardware that can map one or more logical tasks on a single PE.

According to current knowledge this is the first work in which the use of simulated annealing has been applied in co-synthesis embedded systems considering divisibility of tasks as attribute tasks.

4 Simulated Annealing

It belongs to the group of so-called soft selection algorithms. These are algorithms which allow to accept the growth of minimized objective function, in order to avoid local minima zone. The basis for this algorithm is an analogy of creating a crystal solid through the slow process of annealing. The aim is to achieve a minimum energy state corresponding to crystal. This is accomplished by the slow decreasing of temperature, so that at each temperature level heat balance is obtained.

Because the scheduling is classified as NP-hard problems still are being sought as the most optimum solutions using algorithms approximating the most accurate results. The best solution to the problem can be found by reviewing all possible mappings, which in a multiprocessor system based on a network of NoC is first brought to the mapping tasks in the correct network and then scheduling at the appropriate processors tasks and the required transmission. This method is however impractical due to exponential number of possible solutions.

One example solves the problem of mapping and scheduling the network NoC is the method of simulated annealing [15–17]. This is an example of a method of performing global optimization. The idea of this algorithm is taken from the metallurgical where it describes the process of annealing of metals.

Initially, the algorithm tends to accept many random solutions with all available space as to minimize temperature narrows the space of solutions until the freeze. Stopping criterion is determined by the freezing condition or lack of changes by a fixed number of iterations. In this work simulated annealing algorithm was used to find the most optimal solution.

5 Preliminaries

This paper presents a co-synthesis hardware and software method based on the simulated annealing method [11, 12]. The comparison of the methods used synthesis hardware and software for the tasks and problems divisible and indivisible is shown here. So far, the proposed approach to co-synthesis using any methods or algorithms not considered such an attribute which is the divisibility of tasks. According to the current state of knowledge of the author it has not been studied as such issues in terms of scheduling. The approach presented here is also proposing to split the graph into individual regions, also performing at the minimization of certain parameters of NoC network.

Acyclic directed graph can be used to describe the proposed embedded system. Each node in the graph represents a task, while the edge describes the relationship between related tasks. Each of the edges in the graph is marked by a label d, where each of its indices defines the tasks that connect. An exemplary task graph is shown in Fig. 1. In the presented approach time constraints on the chosen path in the graph are taken into account. In Fig. 1 presents the restrictions (deadlines) imposed on the path, are marked them as T (rest).

The presented approach assumes that there is a database containing computing elements for implementation in SoC systems:

- tasks execution times
- modules surface

In the proposed method two basic types of PE can be distinguished:

- Universal programmable processor (PP)
- Specialized hardware modules (HC)

Each of universal programmable processors can perform all the tasks that are compatible with them. Ppi is determined by memory space occupied by the required task, while SPP_I determines the area of PP_i processor.

In contrast, PP_i, HC hardware module can perform only one task, but there is a possibility of many hardware implementations of the same task. In addition, certain communication buses are also defined by channels of communication bandwidth. All communication channels have the same bandwidth b_{NOC}.

6 Methodology

The graph showing task dependencies in the constructed system is accepted as input data in the proposed methodology for the simulated annealing. Parts of the graph are submitted to annealing process, in accordance with the proposed optimization. Then the individual parts are combined with each other at the same time specifying the best possible architecture of the system.

6.1 System Design Futures

The proposed system includes the following functionalities:

- graph distribution into relevant regions
- mapping of certain regions into NoC
- annealing process which individual parts of the graph are subjected to

Each of the regions is subjected to annealing process in such a way as to meet the limits and get the best possible reproduction into NoC or NoC region.

System construction consists in the following steps:

1. Drawing a number of algorithms for partitioning and mapping
2. Drawing division algorithms from the pool of available algorithms
3. Graph division in accordance with the rules of algorithm drawn
4. Drawing the calibration algorithms from the pool of available algorithms
5. Representation of input task graph part in the created portion of NoC
6. The combination of all the NoC regions into one network.

After mapping all tasks in the appropriate regions of NoC, one integral network has created algorithm set for divisibility and mapping together with a description of the values optimized by particular rules are presented below. Each of the proposed algorithms suggests that one or more of the parameters associated with NoC problems should be minimized. The above table (Table 1) shows selected individual NoC parameters proposed for optimization, such as: optimizing the speed, optimizing the NoC traffic or optimizing NoC energy consumption. Optimizing NoC traffic is carried out by the division gene proposing a cut at the point of the longest transmission from the graph which is being examined. Removal of the specified transmission adds a communication channel between the relevant NoC regions. Disabling the longest transmission from the available NoC region provides more options for aligning the remaining transmissions.

Table 1. Sample database for the system described by the Fig. 1

Task name	P1		P2		P3	
	t	s	t	s	t	s
T1	15	5	t	s	41	13
T2	15	13	37	15	28	21
T3	15	21	26	23	20	23
T4	10	20	23	25	20	38
T5	10	28	18	40	20	35
T6	10	7	11	38	20	12
T7	10	15	26	15	20	19
T8	10	23	27	22	20	23
T9	10	18	18	27	20	35
T10	10	30	11	38	20	37
T11	10	11	12	41	20	26
T12	15	17	42	31	22	34
T13	15	15	15	37	14	21
T14	15	9	16	25	25	13
T15	15	11	15	17	30	8
T16	20	16	27	9	35	18
T17	20	18	18	27	22	35
T18	20	30	11	38	23	37
T19	20	11	12	41	51	26

In the proposed approach, there are two types of algorithms to be distinguished: dividing algorithms and mapping algorithms. The first type is responsible for the breakdown of the graph into smaller parts of the network. The second type of algorithm is used for representation of a portion of the graph (which is the result of division by sharing algorithm/algorithms) into NoC region. Then each of the proposed (allocated) graph fragments undergoes the process of finding the best solution for a given NoC region by means of applying the annealing algorithm. After finding a set of solutions for all the regions they are merged into one coherent and integral dedicated network. Figure 1 shows a sample tasks graph with the tasks use as input to creating system. Each of the proposed algorithms must ensure full compliance with the time limits imposed on the graph that represents system tasks. The next step after the division task to processors is task-specific mapping step in the specified region of NoC. The next step is to create complete network architecture, combining all its regions and, if necessary, regrouping tasks, as shown in Fig. 2. Architecture for the tasks usually attribute divisible without it differ from one another, but in this case they are the same and are shown in Fig. 2. Last step is scheduling. This is shown in Fig. 3 (for non attribute of divisible tasks) and Fig. 4 (for the task with attribute of divisibility).

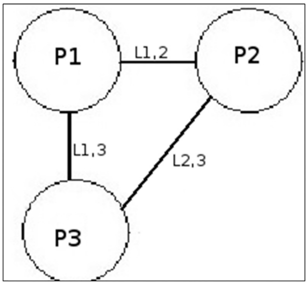


Fig. 2. NoC architecture for indivisibility and divisibility tasks

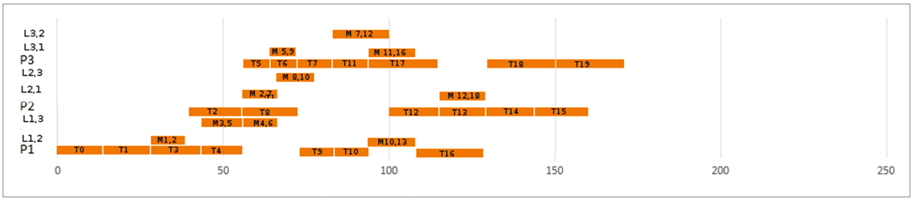


Fig. 3. Scheduling of invisibility tasks

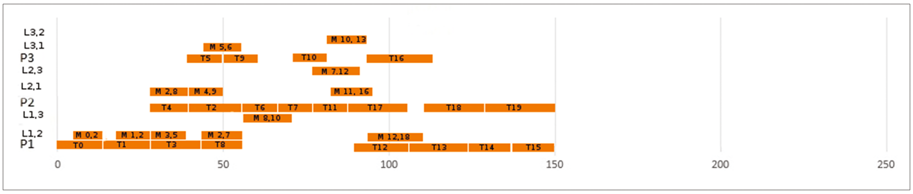


Fig. 4. Scheduling of divisibility tasks

7 Result and Analysis

Research has been carried out on comparing task allocation with regard to divisibility attribute and without taking account of this attribute. The same algorithms were used for each of the cases under examination. The application of the proposed approach to solving the problem resulted in a significant improvement in system performance as well as in the implementation of all tasks and allowed for lower power consumption. Division of graph into individual regions, which are then mapped into parts of NoC, which was already proposed, brings significant benefits in relation to representation of the graph as a whole.

Because previous studies have been carried out on the graphs generated by the author's work in the current was used commonly known graphs presented for open discussion and testing with their use. For graphs used in the paper, the author applied for divisibility attribute randomly selected tasks. What is a new approach in relation to the proposed works [2, 3]. Like the previous studies use divisibility attribute importance shortened the duration of the system. The following are generated to the target architecture with all the data needed to create the system. Research has shown that by using the attribute divisible of tasks, the obtained execution time is shorter than the algorithm proposed by authors of TGFF graphs. The problem turned out to scheduling tasks for some graphs TGFF without the use of an attribute divisibility. Thanks to the additional use of task divisibility attribute, much faster execution of the system was possible, as well as further energy was saved thanks to the proposed solution.

For the cases referred to were obtained times worse than those obtained by the authors of graphs, which motivates you to change the algorithm search space of possible solutions to thereby further shorten the generation of the target system.

8 Summary

This work considers the problem of scheduling tasks in embedded systems including attribute divisibility. The envisaged system is specified using the graph tasks. The target system is constructed using the algorithm of simulated expression. In the process of constructing the system and actually improve the original solution process is carried out scheduling and, if necessary, the allocation of tasks to other processors to meet the criteria of multi-criteria optimization in the process of creating a new system. The most important criterion in the process of creating the system is to meet the restrictions imposed on individual path in the graph representing the system tasks to create. The presented approach is an extension of the work presented in the article "Scheduling tasks in embedded systems based on NoC architecture, International Journal of Computer Technology and Applications, 2014". As in the case of said work studies, in researches the same algorithms was used for each case considered. Previous studies have indicated very promising results after applying the attribute of divisibility for tasks represented in the created systems.

The study allowed us to achieve shorter lead times prioritization of tasks for graphs TGFF than those obtained by the authors. In future work would be an interesting issue to consider dual-processor tasks attribute divisibility and without this attribute.

References

1. Dorota, D.: Scheduling tasks in embedded systems based on NoC architecture. Int. J. Comput. Technol. Appl. **5**, 1909–1916 (2014)
2. Dick, R.P., Rhodes, D.L., Wolf, W.: TGFF: task graphs for free. In: Proceedings of the 6th International Workshop on Hardware/Software Codesign. IEEE Computer Society (1998)
3. Dick, R., Rhodes, D.: TGFF. http://ziyang.eecs.umich.edu/~dickrp/tgff/. (Accessed Dec 2016)
4. Kopetz, H.: Real-time Systems: Design Principles for Distributed Embedded Applications. Springer (2011)
5. Lee, E.A., Seshia, S.A.: Introduction to Embedded Systems: A Cyber-physical Systems Approach. Lee & Seshia, Berkeley (2011)
6. Rajesh, K.G.: Co-synthesis of hardware and software for digital embedded systems, Ph.D. thesis, 10 December 1993
7. Ost, L., Mandelli, M., Almeida, G.M., Moller, L., Indrusiak, L.S., Sassatelli, G., Moraes, F.: Power-aware dynamic mapping heuristics for NoC-based MPSoCs using a unified model-based approach. ACM Trans. Embed. Comput. Syst. (TECS) **12**(3), 75 (2013)
8. Oh, H., Ha, S.: Hardware-software cosynthesis of multi-mode multi-task embedded systems with real-time constraints. In: Proceedings of the Tenth International Symposium on Hardware/Software Codesign, pp. 133–138. ACM (2002)
9. Strachacki M.: Projektowanie i optymalizacja sprzętowo-programowych wbudowanych systemów przetwarzania danych, Ph.D. thesis, Gdańsk, February 2008
10. Khan, G.N., Iniewski, K. (eds.).: Embedded and Networking Systems: Design, Software, and Implementation. CRC Press (2013)
11. Eles, P., Peng, Z., Kuchcinski, K., Doboli, A.: System level hardware/software partitioning based on simulated annealing and tabu search. Des. Autom. Embed. Syst. **2**(1), 5–32 (1997)
12. Henkel, J., Ernst, R., Holtmann, U., Benner, T.: Adaptation of partitioning and high-level synthesis in hardware/software co-synthesis. In: Proceedings of the 1994 IEEE/ACM International Conference on Computer-Aided Design, pp. 96–100. IEEE Computer Society Press, November 1994
13. O'Connor, I., Nicolescu, G. (eds.).: Integrated Optical Interconnect Architectures for Embedded Systems. Springer 2013
14. Wettin, P., Murray, J., Kim, R., Yu, X., Pande, P.P., Heo, D.: Performance evaluation of wireless NoCs in presence of irregular network routing strategies. In: Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 1–6. IEEE (2014)
15. Dick, R.P., Jha, N.K.: MOGAC: a multiobjective genetic algorithm for hardware-software cosynthesis of distributed embedded systems. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **17**(10), 920–935 (1998)
16. Chinneck, J.W.: Practical Optimization: a Gentle Introduction, Lecture Notes, Systems and Computer Engineering, Ph.D. thesis, Carleton University, Ottawa, Canada, 12 December 2010
17. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Sci. New Ser. **220**(4598), 671–680 (1983)

Adaptation of Ant Colony Algorithm for CAD of Complex Systems with Higher Degree of Dependability

Mieczyslaw Drabowski(✉)

Cracow University of Technology, Warszawska 24, 31-155 Kraków, Poland
drabowski@pk.edu.pl

Abstract. The paper includes a proposal of a new algorithm for Computer Aided Design (CAD) of complex system with higher degree of dependability. Optimization: in scheduling of tasks, partitioning of resources, the allocation of task and resources are basic goals this algorithm. These optimization problems are NP-hard, but can be it solved efficiently e.g. by meta-heuristic algorithms. Presented the CAD algorithm based on Ant Colony Optimization may have a practical application in developing tools for rapid prototyping of such systems. The Ant Colony Optimization algorithm is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs.

Keywords: Complex system · Scheduling · Partition · Allocation · Dependable · Optimization · Pheromone · Evaporation · Ant Colony · CAD tools

1 Introduction

The objective of computer-aided design of complex computer systems (i.e. that contain sets of processors and additional resources, that compute great number of programs) is to find the optimal solution, in accordance with the requirements and constraints imposed by of the stated specification of the systems. The following optimality criteria are usually taken into consideration: the speed of action, the cost of the implementation, the power of consumption and the degree of dependability.

The identification and partitioning of resources between various implementation techniques is the basic matter of automatic aided design. Such partitioning is significant, because every complex system must be realized as result of hardware implementation for its certain tasks and of software implementation for other. Additionally scheduling problems are one of the most significant issues occurring in design of operating procedures responsible for controlling the allocations of tasks and resources in complex systems.

The new model and new construction methods - the par-synthesis - which are presented in [1], software and hardware components are developed jointly in parallel and coherently connected to each other, what the final solution reduced costs and increases the speed of action, differently than before. This model and methods for systems with a high degree of dependability were presented in [2].

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_14

The resources distribution is to specify, what hardware and software are in system and to allocate theirs to specific tasks, before designing execution details. Another important issue that occurs in designing complex systems is assuring their fault-free operation. Such designing concentrates on developing dependable and fault-tolerant architectures and constructing dedicated operating procedures for them [3]. In this system an appropriate strategy of self-testing during regular exploitation must be provided. The general model and new concept of parallel to tasks scheduling and resources partition for complex systems with higher degree of dependability was presented in detail in [3, 4]. We proposed in these papers the following schematic diagram of a coherent process of synthesis for systems of faults tolerant, Fig. 1, which in turn increases the degree of dependability of these systems.

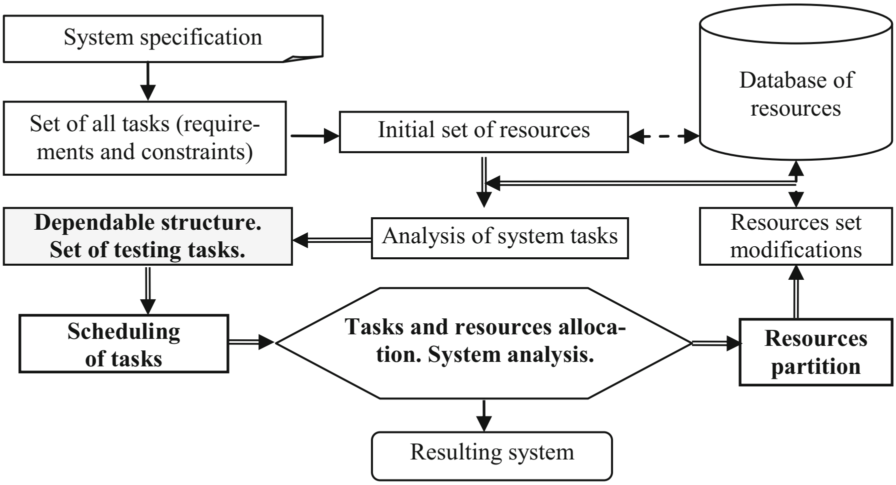

System specification
Set of all tasks (requirements and constraints)
Initial set of resources
Database of resources
Dependable structure. Set of testing tasks.
Analysis of system tasks
Resources set modifications
Scheduling of tasks
Tasks and resources allocation. System analysis.
Resources partition
Resulting system

Fig. 1. The process par-synthesis of dependable complex system

The suggested method of par-synthesis consists of the following steps:

1. specification of requirements and constraints for the system,
2. specification of all tasks,
3. assuming the initial values of resource set,
4. defining testing tasks and the structure of system, self-testing strategy selection,
5. scheduling of tasks,
6. the evaluation the operating speed and system cost, multi-criteria optimization,
7. the change should be followed by a modification of the resource set, a new system partitioning into hardware and software parts and an update of structure and test tasks (go to step 5).

Modeling fault tolerant systems consists of resource identification and tasks scheduling problems that are both hard NP-complete [5, 6]. Algorithms for solving such problems are usually based on heuristic approaches.

The objective of this paper is to present of meta-heuristic approaches to the solution problems of dependable complex systems design, i.e. a simultaneously solution to tasks scheduling and resource assignment problems. The set of tasks consists with tasks of self testing and tasks users. A collection of resources creates the redundant structure with high dependability. We suggested in the paper [3] hybrid algorithm: evolutionary with simulated annealing, in which there are Boltzmann tournaments. The paper [4] presented other the meta-heuristic algorithm, based for the adaptation a neural network Tsang-Wang, what has been applied for resolving problems of resource allocation and task scheduling and also for a coherent solution to these problems. We present, in this paper, other the meta-heuristic algorithm yet, based for the adaptation Ant Colony Optimization, for aided design of system of higher degree of dependability. The development of these algorithms allowed us to compare the results achieved by these various meta-heuristics.

2 Adaptation of ACO to Solve the Problems of Par-Synthesis

The Ant Colony Optimization (ACO) algorithm is a heuristic algorithm which uses the idea of agents (here: ants) and imitating their real behavior in a way [7, 8]. On the basis of specific information (distance, amount of pheromone on the paths, etc.) ants evaluate the quality of paths and choose between them with some random probability (the better path quality, the higher probability it represents). Having walked the whole path from the source to destination, ants learn from one another by leaving a layer of pheromone on the path. Its amount depends on the quality of solution chosen by agent: the better solution, the bigger is amount of pheromone. Then the pheromone "vapors" to enable the change of path chosen by ants and let them ignore the worse (more distant from targets) paths, which they were walking earlier. The result of such an algorithmic operation is not only the solution itself. Very often it is the trace, which leads to this solution. It allows us to analyze not only a single solution, but also permutations generating different solutions, although based on the same division (i.e. tasks are scheduled in different order, though they are still allocated to the same processors). This very kind of approach is utilized to solve problems such as synthesis, where not only the partition of tasks is important, but also their schedule.

To adapt the ACO algorithm to synthesis problems, the following two parameters have been defined:

- Number of agents (ants) in the colony.
- Pheromone evaporation coefficient (from the range (0; 1)).

The process of choosing these parameters is important and should consider that:

- For too big number of agents, the individual cycle of algorithm can last quite long, and the values saved in the table ("levels of pheromone") as a result of addition will determine relatively weak solutions.
- On the other hand, when the number of agents is too small, most of paths will not be covered and as a result, the best solution can long be uncovered.

The situation is similar for the evaporation coefficient:

- Too small value will cause that ants will quickly “forget” good solutions and as a result it can quickly come to the so called *stagnation* (the algorithm will stop at one solution, which does not have to be the best one).
- Too big value of this factor will make ants don’t stop analyze “weak” solutions; furthermore, the new solutions may not be pushed, if time, which has passed since the last solution found will be long enough (it is the values of pheromone saved in the table will be too big).

The ACO algorithm defines two more parameters, which let you balance between:

- α – the amount of pheromone on the path, and
- β – “quality” of the next step.

These parameters are chosen for specific instance of problem. This way, for parameters:

- $\alpha > \beta$ there is bigger influence on the choice of path, which is more often exploited,
- $\alpha < \beta$ there is bigger influence on the choice of path, which offers better solution,
- $\alpha = \beta$ there is balanced dependency between quality of the path and degree of its exploitation,
- $\alpha = 0$ there is a heuristic algorithm based only on the quality of passage between consecutive points (ignorance of the pheromone level on the path),
- $\beta = 0$ there is a heuristic algorithm based only on the amount of pheromone (it is the factor of path attendance),
- $\alpha = \beta = 0$ we’ll get the algorithm making division evenly and independently of the amount of pheromone or the quality of solution.

Having given the set of neighborhood $\boldsymbol{N}$ of the given point $\boldsymbol{i}$, amount of pheromone on the path $\boldsymbol{\tau}$ and the quality of passage from point $\boldsymbol{i}$ to point $\boldsymbol{j}$ as an element of the table $\boldsymbol{\eta}$ you can present the probability of passage from point $\boldsymbol{i}$ to $\boldsymbol{j}$ as [9]:

$$p_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum\limits_{l \in N_l^k} [\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}} & \text{when j} \in N_i^k \\ 0 & \text{else} \end{cases}$$

Formula 2.1. Evaluation of the quality of the next step in the ACO algorithm.

In the approach presented here, the ACO algorithm uses agents to find three pieces of information [9, 10]:

- The best/the most beneficial division of tasks between processors.
- The best sequence of tasks.
- Searching for the best possible solution for the given distribution.

Agents (ants) are searching for the solutions which are the collection resulting from the first two targets (they give the unique solution as a result). After scheduling, agents fill in two tables:

- Two-dimensional table representing allocation of task to the given processor.
- One-dimensional table representing the sequence of running the tasks.

The operation of agent involves (Fig. 2).

To evaluate the quality of task allocation to processor, the following method is being used (Fig. 3).

The computational complexity of single agent is polynomial and depends on the number of tasks, resources and times of tasks beginning.

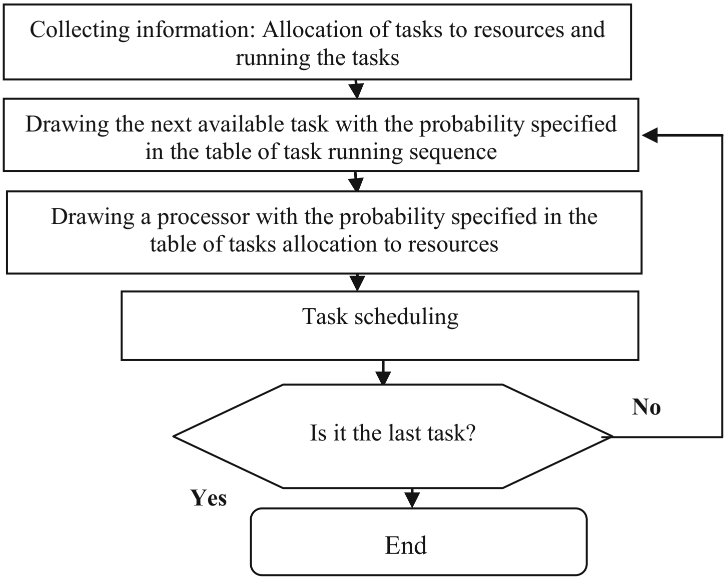

Collecting information: Allocation of tasks to resources and running the tasks
Drawing the next available task with the probability specified in the table of task running sequence
Drawing a processor with the probability specified in the table of tasks allocation to resources
Task scheduling
Is it the last task?
No
Yes
End

Fig. 2. Agent (ant) operation scheme

Selected algorithm parameters:

- ***a*** is number of ants: for number of tasks ***n < 50*** this ***a = 75*** and for ***n >= 50*** this ***a = [1,5 • n]*** i.e. the largest integer not greater than ***1,5 • n***
- the pheromone evaporation coefficient ***is 0,08.***

After initiating the tables (of allocation and sequence) for each agent, the algorithm starts the above cycle, after which the evaluation of solutions takes place. Having completed the particular number of cycles, the parameters are being updated and algorithm continues working (Fig. 4).

Like in other meta-heuristic algorithms, parameters of ants' colony have been selected through experiments. Algorithm tuning is to select possibly best parameter values. This process demands many experiments which are conducted for different

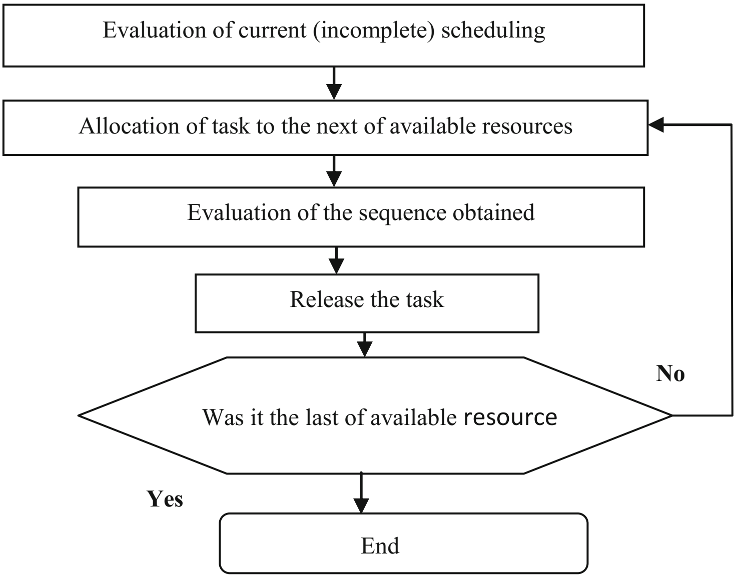

Evaluation of current (incomplete) scheduling
Allocation of task to the next of available resources
Evaluation of the sequence obtained
Release the task
Was it the last of available resource
No
Yes
End

Fig. 3. The principle of path evaluation

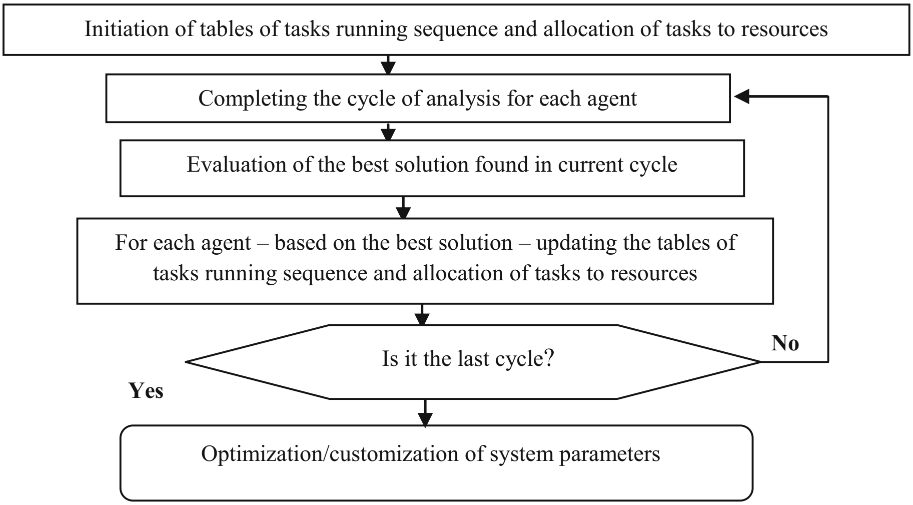

Initiation of tables of tasks running sequence and allocation of tasks to resources
Completing the cycle of analysis for each agent
Evaluation of the best solution found in current cycle
For each agent – based on the best solution – updating the tables of tasks running sequence and allocation of tasks to resources
Is it the last cycle?
No
Yes
Optimization/customization of system parameters

Fig. 4. The principle of ACO algorithm operation

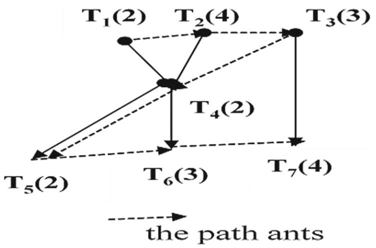
combinations of parameter values. For each combination of variable values, computation process has been repeated many times, and then an average result has been calculated. The same graphs type of STG [11], like at previous algorithms (Genetic and Neural), have been applied.

3 Computational Experiments

3.1 Example Actions of Algorithm for Scheduling Problem

Specifications: Two identical parallel processors and seven dependent nonpreemptive tasks with known execution times. Should be found allocations and priorities (for $\alpha > \beta$).

1. Define the G - structure of the tasks unallocated and the S - structure of the possible for allocation of tasks in the next step (for example, at the beginning of $G = \{T_1, T_2, \ldots, T_7\}$ and $S = \{T_1, T_2, T_3\}$). Update pheromone level, include evaporation.
2. In the S choose the task with the strongest track structure with pheromone: e.g. after the task T_3 is selected the task T_4, and not T_7, if dominated by the quality of the path: e.g. the length available next task, would be selected the task T_7.
3. Allocate available the task to the processor - just as soon as you can and in accordance with the precedence constraints.
4. Remove the selected task of G and S, and add to the list of correction by an ant.
5. Update the level of pheromone and leave a trace (Fig. 5).
6. If G = Ø is the end of the algorithm.
7. Go to point 1;



processor	*P1*	T	1		T3			T	5		T	7	
processor	*P2*		T	2		T	4		T	6			
time	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*

Fig. 5. Gantt chart the result of scheduling

3.2 Examples of Selected Computational Experiments. Comparing the Results of Par- Synthesis for Dependable Systems

3.2.1 Independent Tasks with Cost of Memory – Minimization of Cost and Time

Algorithm Ant Colony Optimization obtained solutions with lower cost than other algorithms solutions for almost all instances of problems. Solutions obtained with this algorithms feature shorter scheduling lengths when compared to solutions of other

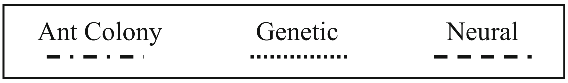


Fig. 6. The markings on the charts for algorithms: ACO, Genetic [4], Neural [5], respectively.

algorithms. When the number of tasks is increased, ACO algorithm outruns others by generating structures with much shorter scheduling lengths – Charts 1 and 2 (Fig. 6).

3.2.2 The Influence of Precedence Constraints on Computations

Calculations were conducted for ten sets of dependent, nonpreemptable tasks, with different numbers of tasks and sequence constraints. Maximal number of processors: 15. Both time and cost were optimization criteria. Random access memory cost was taken into account as well.

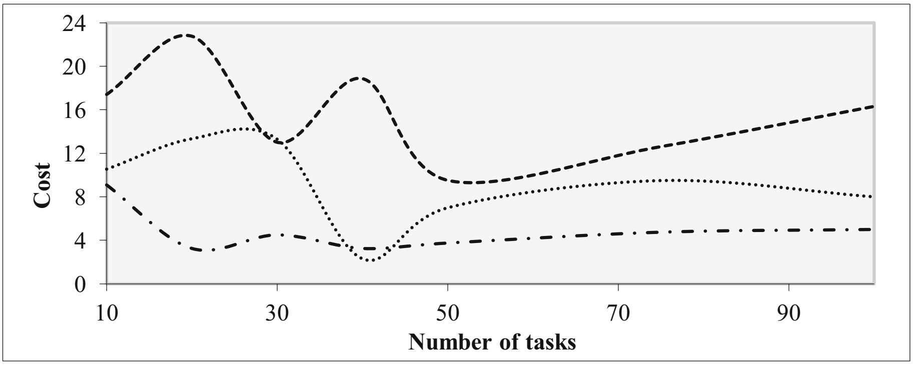


Chart 1. Minimization of cost

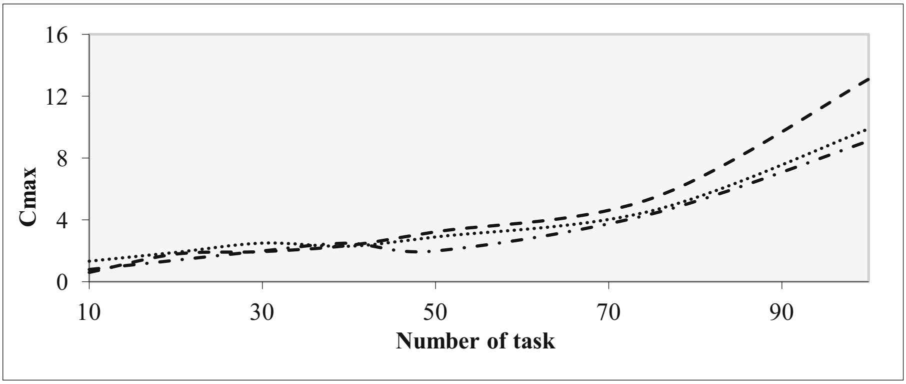


Chart 2. Minimization of time

Once again ACO algorithm obtained solutions at a lower cost than solutions of other algorithms. The case is different for time criterion (scheduling length) – there are instances of problems where similar and better solutions are obtained by genetic algorithm, especially for the criterion on time – Charts 3 and 4.

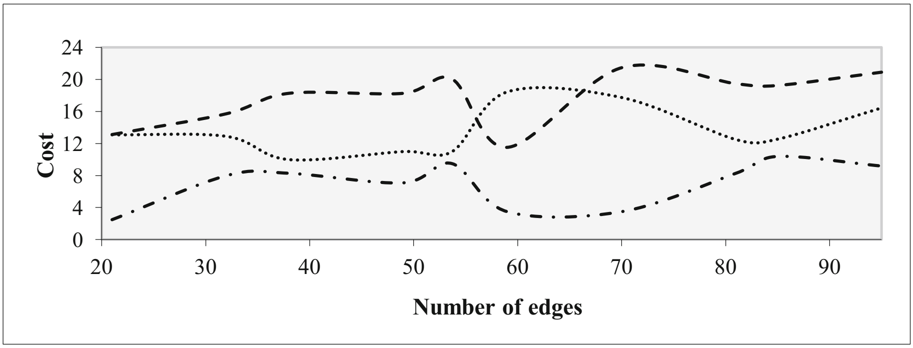


Chart 3. Influence of precedence constraints on cost

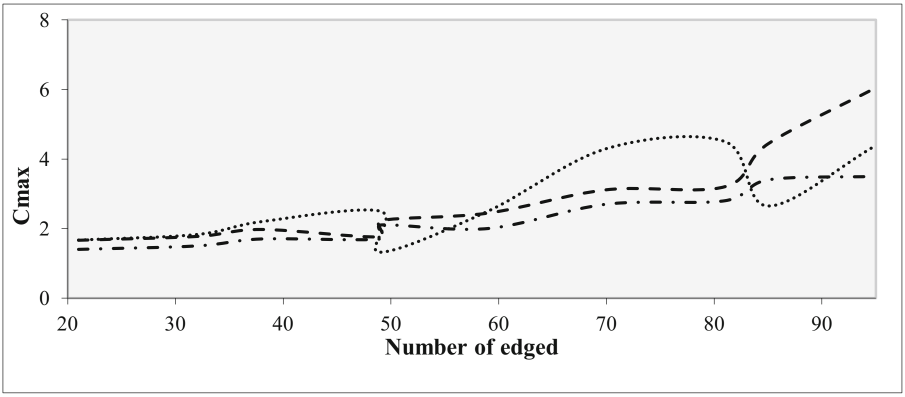


Chart 4. Influence of precedence constraints on time

4 Conclusions

This paper is about the problems of parallel design of complex systems with high degree of dependability. Such a design is carried out on a high level of abstraction in which system specification consists of a set of tasks, which should be implemented by a set of resources and these are listed in the database (container or a catalogues) and are available (exist or can be fast created). Resources possess certain constraints and characteristics, including speed, power, cost and dependability parameters. Thus such a design concerns complex system of the following type – resources, tasks and optimization criterions. The problems of resource partitioning (selection) as well as scheduling (sequencing) of tasks performed on these resources are determined simultaneously. Optimization of afore-mentioned design actions occurs on the same high level.

Among presented in this paper results of computational experiments the new solutions were obtained with method Ant Colony Optimization.

In papers [3, 4] were presented other methods - Genetic i Neural - was possible to compare the results for of various methods, which selected are presented here.

For different problem instances, particular algorithms may achieve different successes; others may achieve worse or better results at different numbers of tasks or for other optimization criterions.

The recommended option is to obtain of results by different algorithms for different their actions and the compare these solutions.

Adaptations of other methods and additional comparison of the results obtained will now be studied.

References

1. Drabowski, M.: Par-synthesis of multiprocessors parallel systems. Int. J. Comput. Sci. Netw. Secur. **8**(10), 90–96 (2008)
2. Drabowski, M., Wantuch, E.: Deterministic schedule of task in multiprocessor computer systems with higher degree of dependability. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCos-RELCOMEX. Advances in Intelligent Systems and Computing, vol. 286, pp. 165–175. Springer (2014)
3. Drabowski, M.: Boltzmann tournaments in evolutionary algorithm for CAD of complex systems with higher degree of dependability. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Tenth International Conference on Dependability and Complex Systems DepCos-RELCOMEX. Advances in Intelligent Systems and Computing, vol. 365, pp. 141–152. Springer (2015)
4. Drabowski, M.: Modification of neural network Tsang-Wang in algorithm for CAD of systems with higher degree of dependability. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) 11th International Conference on Dependability and Complex Systems DepCos-RELCOMEX. Advances in Intelligent Systems and Computing, vol. 470, pp. 121–133. Springer (2016)
5. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
6. Błażewicz, J., Drabowski, M., Węglarz, J.: Scheduling multiprocessor tasks to minimize schedule length. IEEE Trans. Comput. **C-35**(5), 389–393 (1986)
7. Dorigo, M., Di Caro, G., Gambardella, L.M.: An algorithms for discrete optimization. Artif. Life **5**(2), 137–172 (1999)
8. Blum, C.: Beam-ACO – hybridizing ant colony optimization with bean search: an application to open shop scheduling. Comput. Oper. Res. **32**, 1565–1591 (2005)
9. Blum, C., Sampels, M.: An ant colony optimization algorithm for shop scheduling problems. J. Math. Model. Algorithm **3**, 285–308 (2004)
10. Montgomery, J., Fayad, C., Petrovic, S.: Solution representation for job shop scheduling problems in ant colony optimization. LNCS **4150**, 484–491 (2006)
11. http://www.kasahara.elec.waseda.ac.jp/schedule/stgarc_e.html

Context-Aware Anomaly Detection in Embedded Systems

Fatemeh Ehsani-Besheli(✉) and Hamid R. Zarandi

Department of Computer Engineering and Information Technology,
Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
{ehhsani,h_zarandi}@aut.ac.ir

Abstract. To meet the reliability of embedded systems, fault-tolerant methods are widely used. The first step in many of these methods is detecting faults and anomaly detection is often the primary technique which leads to early indication of faults. In the context of embedded systems, some anomaly detection methods are available however; none of them are adaptable to dynamic environments. All of the previous works attempt to provide anomaly detection systems without considering the context of the data. Contextual anomalies, also referred to as conditional anomalies, have different behavior in different contexts. The purpose of designing a context-aware anomaly detection mechanism is to provide the capability of detecting anomalies while the system's environment changes. In this paper, a method for detecting anomalies is proposed which adapts itself to the changes in dynamic environments during detection phase. This method first gives the context of a small window in a data flow and then loads corresponding configuration to the anomaly detector. The results have shown an average of 68.83% of true positive rate and 11.41% of false alarm rate.

Keywords: Anomaly · Context-aware anomaly detection · Categorical data · Dependability · Embedded systems

1 Introduction

Embedded systems have become inevitable parts of a diverse range of systems. They are being deployed in many application areas ranging from control of critical infrastructures to catch data from hostile environments [1]. Many of these devices with embedded computers are safety critical and any malfunction can cause damages or loss in human life and property, and therefore will require a higher level of dependability than usual. It is presumed that fault tolerance, which is one way of achieving high dependability, will be employed in such devices. In several fault-tolerant mechanisms, the first step requires fault to be detected and then, fault tolerance measures can be applied to tolerate it. Hence, fault detection is a primary step in achieving dependability [2].

Anomaly is an event or pattern in data that differs from a standard notion of normal behavior. An anomalous event deflection from the expected behavior is higher than a pre-defined threshold [3]. Threshold is denoted as the lowest level of deviation from normalcy. According to the distance metric, any event exceeding the threshold is beyond the normal region and considered anomalous. Anomaly detection refers to the

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_15

problem of finding patterns in data which deviate substantially from a known behavior [4]. The extensive use of anomaly detection is in a wide variety of application domains such as fraud detection for credit cards, intrusion detection for security monitoring systems, fault detection in military surveillance, safety critical systems and in many other areas. The importance of anomaly detection is because of translating anomalies of data into important (and often critical) practicable information in an extensive application domain.

Fault detection can be performed either explicitly or implicitly [2, 4]. In the former, faults are typically detected through pattern recognition, which classifies data based on a priori knowledge or by extracting statistical information from the patterns [5]. In the latter, faults are detected using some indicators, such as anomaly, which may have been caused by faults. Some faults do not have any distinct signature and are detected indirectly as a result of unusual behavior of the system that manifest itself as anomalies in sensor data or monitored systems [6]. The data collected by such sensors are called the sensor data that can be numeric or categorical. Numerical data often referred to as the measurable data, are usually continuous have a unique zero point on a ratio scale and have mathematical ordering properties. Categorical data, also called qualitative or nominal data, are counts for a set of non-overlapping categories. They have no absolute measurements, cannot be ordered by their values, but can be separated into groups [7, 8]. It seems likely that with increasing computing power, more of the sensor data will be in the form of categorical data [2]. Working with categorical data, since they are much more difficult to handle than numerical data, is a real challenge to users and developers of such sensors [2].

Categorical data sets contain sequences of symbols. A collection of unique symbols makes the alphabet. An anomalous event may occur when some particular symbols or subsequences are adjacently located in the data stream. When system's behavior is considered normal for the environment, the data collected in the monitored sensors used as the training dataset. During the collection period, which is called the training phase, it is required that no unusual conditions dominate the operation of system [2].

Anomalies can be divided into three main types: (1) Point, (2) Collective, and (3) Contextual anomalies. Point anomalies occur when an individual data instance is anomalous with respect to w.r.t all other data in the dataset. This is the most common anomaly discussed in literature. Collective anomalies occur when a data instance is anomalous based on the values of its adjacent items. This is prevalent in sequence data, graph data, and spatial data. The final type, contextual or conditional anomalies occur when the data instance is only anomalous in a specific context. Most research is concerned with the two prior types of anomalies with less focuses on the contextual anomalies [3, 9].

This paper deals with the problem of detecting anomalies in a non-stationary environment with categorical data. Considering that, normal behaviors may change in different circumstances due to changes in some environmental factors [10], there is a need for a detection mechanism which is adaptive to the changes during its detection phase. Our anomaly detection technique is a window-based method and consists of three steps: (1) Training in design time, (2) Identifying the data context in run time, (3) Detecting anomaly in run time.

Training in design time: Probability matrices are calculated. Each probability matrix shows the probability of each unique subsequence in the normal data. Moreover representative vectors are generated. These are explained more in the following sections.

Identifying the data context in run time: Data context is recognized by k-Nearest Neighbor (*kNN*) classifier which assigns a class label to the captured sequence of the test data using the representatives in the learning set.

Detecting anomaly in run time: When the context is recognized, the algorithm automatically assigns the correct probability matrix for the context, and anomaly is detected based on the Probabilistic method.

Major metrics for evaluating the proposed method are detection coverage, power, area and time consumptions. The evaluation was done using a total of 460 test data sets for different number of injected anomalies. The range of injected anomalies varies between one to six. The average of 68.83% of true positive and 11.41% of false negative rate shows the effectiveness of the proposed method on improving fault-tolerance metrics of the embedded systems.

The rest of the paper is organized as follows: Sect. 2 gives the related work. Section 3 presents the overview of our anomaly detection scheme. Section 4 describes the construction of the benchmark datasets. Section 5 carries out the hardware implementation. Section 6 discusses the experimental results and, finally, Sect. 7 draws conclusions.

2 Backgrounds and Related Work

This section presents the terminology and the definitions related to the proposed method.

Anomalous Event: An anomalous event in the output of the sensor is the one consist of symbol or permutation of symbols which do not exist in the normal data set. There are three phenomena which could make an event anomalous [2]:

Foreign symbol: These anomalies occur when a symbol not included in the training data set, comes in testing data set.

Foreign sequence: A set of ordered elements containing no foreign symbol that is not present in the training data set.

Rare sequences: Sequences which occurs less than a predefined threshold in the training data set.

An anomaly detector should be capable to make decision about events within its scope. It cannot decide on the events it cannot see. The scope of an anomaly detector is the extent which the detector window overlaps the anomaly called as [2]: the *Whole Scope* (detector window size equal to anomaly length), *Internal Scope* (detector window size less than anomaly length), *Encompassing Scope* (detector window size larger than anomaly length), *Boundary Scope* (detector see a part of the anomaly as well as part of the background data), *Background Scope* (detector window see only the background data).

2.1 Anomaly Detection Techniques

Anomaly detection has been a focus of many research papers within diverse areas. In network area, the naive Bayesian classification method has been used to detect anomalies in the sensor data [11, 12]. Several methods have been developed for detecting anomalies in operating system call data [13]. A set of anomaly detection techniques has been proposed that detect anomalies in large flight datasets.

Based on what is considered as a unit element in a test sequence, the anomaly detection techniques are classified into four categories including: Kernel-based, Markovian, window-based and Hidden Markov Models techniques. Kernel-based techniques are similar to point-based anomaly detection, since the entire test sequence is considered as a unit element. Markovian techniques learn a probabilistic model in the training phase and use this model to predict the occurrence probability of each symbol in the test sequence with respect to the previous few symbols observed so far [2]. In window-based techniques, a short window is moved through the sequence and the symbols within the window are analyzed. Based on the analysis on the subsequence, the anomaly score of the entire test sequences is determined. Hidden Markov Models techniques transform the observable set of symbols to hidden state space using probabilistic rules and detect anomalies in this space.

2.2 Detectors

In the following we give brief reviews of methods that have been presented particularly in the context of anomaly detection in embedded systems.

Markov, STIDE and Probability-based anomaly detectors are some effective methods that have been evaluated in embedded systems.

Markov detector: A Markov model consists of a set of finite states and the probabilities of transiting between them. Figure 1 shows a transitional matrix of a Markov model with four states in which each letter represents a state and numbers are the transitional probabilities between states. The main feature of a Markov model is that the next state (S_{t+1}) in this model only depends on the current state (S_t) not to any previous states. This feature can be written as follows: The state of a system at time t is represented by X_t, then

$$P(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}, \ldots, X_0 = x_0) = P(X_{t+1} = x_{t+1} | X_t = x_t) \quad (1)$$

The Markov anomaly detector calculates the transitional probabilities between states in the training phase and uses them to estimate the transition between states in the test data. The three required steps in a Markov model based anomaly detector are as follows: (1) Constructing a transition matrix from training set, (2) Setting a threshold for the bound within which system's operation is considered normal, (3) Examining the test data using the transition matrix, made in the training phase.

STIDE detector [2]: STIDE (sequence time delay embedding) anomaly detector stores all sequences of a certain size from normal data stream as "self" (normal) sequences and compares the incoming sequence from test data to each of them. If a sequence in the test data does not match any normal sequence, it is identified as "nonself" (anomalous) sequence. The STIDE algorithm can be written as follows:

Let $X = (x_1, x_2, x_3, \ldots, x_N)$ and $Y = (y_1, y_2, y_3, \ldots, y_N)$ be two sequences of length N; the similarity between them is defined by the $Sim(X, Y)$ function:

$$Sim(X,Y) = \begin{cases} 0 & if\, x_i = y_i for\, all\, i,\ 1 \le i \le N \\ 1 & otherwise \end{cases} \tag{2}$$

It states that the *Sim* function returns zero, if the elements of two sequences are peer to peer identical. Given the normal database with M subsequences of size N, $S_n = \{X_1, X_2, X_3, \ldots, X_M\}$ and the test data with K subsequences of size N, $S_q = \{Y_1, Y_2, Y_3, \ldots, Y_K\}$ the anomaly score of a test subsequence Y_t is computed as:

$$A_{Y_t} = \begin{cases} 0 & if\, Sim(X_i, Y_t) = 1 for\, all\, i,\ 1 \le i \le M \\ 1 & otherwise \end{cases} \tag{3}$$

Thus, if the test subsequence (Y_t) does not match any subsequence of the normal database, $A_{Y_t} = 1$. Finally for obtaining an anomaly score for the test sequence (A_{S_q}), the locality frame count (LFC) technique is used. In this technique, the number of mismatching subsequences in a frame with length L (L is a user defined value) of the test sequence is counted. If this number exceeds the threshold, A_{S_q} is incremented.

Probabilistic Detector: This method proposed in [14] is a window-based technique and measures the probability of events for capturing system behavior during the learning phase. While a sliding window is moved through the learning sequence, for all subsequences with the size of the detector's window, the number of pairs of symbols and their distance from each other is calculated in order to make the frequency function. This information is stored in the form of a matrix which will be used in the testing phase.

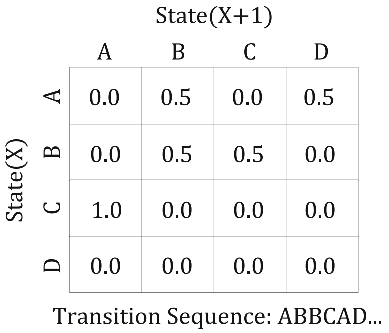
Consider a matrix that is made during the learning stage. Members of the matrix are the frequency function $f_{i(x,y)}$, which represents the number of pairs of elements x and y that occurs at distance i, "$1 < i < (window\, length - 1)$" from each other. Finally, by dividing $f_{i(x,y)}$ to the total number of frequencies of distance "i", an approximate probability of elements x and y at distance "i" is calculated.

In the test stage, the probability of two elements at a specific distance of each other, in the detector's window is obtained from the matrix created in the learning stage and the probability function is defined as:

$$p = \prod\nolimits_{x,y \in DW} p_i(x,y)\, where\, i = d(x,y) \tag{4}$$

Where “DW” is a set of elements in the detection window and the statement $p_i(x, y)$ is the probability of x and y occurring at i distance of each other. The data in the detection window will be more normal, if this product is closer to one.

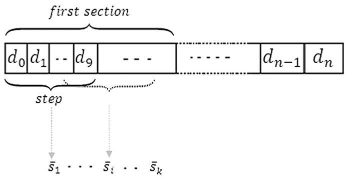
Previous techniques for anomaly detection presented in embedded system, only consider the data itself, without any concern about the context of the data. These methods, assume the environment to be fixed and so they fail to detect anomalies in dynamic environments. An effective anomaly detection method should be capable of recognizing anomalous activities ile adapting to the normal behaviour changes, otherwise the false alarm rate will be greatly increased.



State(X) \ State(X+1)	A	B	C	D
A	0.0	0.5	0.0	0.5
B	0.0	0.5	0.5	0.0
C	1.0	0.0	0.0	0.0
D	0.0	0.0	0.0	0.0

Transition Sequence: ABBCAD...

Fig. 1. Transition matrix for a Markov model with four states. The letters are representing all states and numbers are probabilities between states.



first section
d_0 d_1 $\cdot\cdot$ d_9 - - -
- - - - -
d_{n-1} d_n
step
$\bar{s}_1 \cdot\cdot\cdot \bar{s}_i \cdot\cdot \bar{s}_k$

Fig. 2. Constructing representative vectors for a training sequence

3 Proposing a Context-Aware Anomaly Detection Method for Embedded Systems

The proposed method can be applicable specifically to embedded systems and differs from the detection methods in information theory problems. First, because of the cyber-physical property of embedded systems, the type of data collected from sensors of these systems shows that data does have a correlation with each other. In other word, detection in these systems is performed by considering dependencies in the context of the data. Second, embedded systems substantially must be designed under some constraints such as the limitation on resource usage, time and power consumption and if these constraints are not met, may lead to critical failure or damage.

In this problem, a database of categorical data is collected in which, each single data has two sets of contextual and behavioral attributes. Corresponding to a specific context or condition, the behavior of data may be different.

Data was classified according to their context to different groups or classes, Such that the database is a collection of categorical data in some groups that each corresponds to a specific context.

The proposed anomaly detection method operates in semi-supervised mode where the training strategy will be implemented using only normal instances. The probability matrix presented in the probabilistic detector is also used in both learning and testing stages of this algorithm. Using the probability matrix allows to perform mathematical operations on the data.

3.1 The Training Stage

There is a training sequence in each class that characterizes the normal behavior of data when the system is judged to be under normal usage. In this method, a probability matrix and some representatives are created from the training sequence of each class, and will be used in the test stage. A class representative is a subset of the class members which contains exactly one element from the class.

Consider a database with M classes and assume $(n_i, i \in [1, M])$ representative vectors for each class. The representatives of the i-th class are constructed by dividing the training sequence of that class into n_i sections, and then a predefined number of subsequences with a certain size, shown as *step* in Fig. 2, are randomly selected in each section and the probability matrices are created for them. Eventually, by calculating the average matrix, a representative vector for a section is obtained. To clarify the subject, an example of constructing a representative vector is given next.

Let the training sequence of a class be a stream of n contiguous elements. Also, suppose a set of m representative vectors needs to be defined for that class. Hence, the training sequence is partitioned into m non-overlapping equal length sections. Having set the number of subsequences needed to construct a representative vector to k and the length of each subsequence to q, a representative vector can be generated for one section as explained below.

First, k subsequences are extracted by randomly placing a sliding window of length q over each section elements. Afterwards, the probability matrix is constructed for each subsequence and the average of all the probability matrices is calculated as the representative vector for one section. *The representative vectors of a class are considered as the class members* which will be employed in the test stage. Figure 3 shows an example of a probability matrix and three representatives created in the training stage from a normal sequence.

In (5), $\bar{s}_j$ is the probability matrix of the j-th subsequence extracted from one section of a training sequence and $\bar{R}$ is the average vector of all the probability matrices, indicating the representative vector for that section and is given by,

$$\bar{R} = \frac{\sum_{j=1}^{k} \bar{s}_j}{k} \tag{5}$$

In this function, the distance of a test probability matrix to all classes members (representative vectors) is calculated using a distance measure. Equation (6) shows the Euclidean distance of two vectors $x = (x_{11}, x_{12}, \ldots, x_{nn})$ and r $= (r_{11}, r_{12}, \ldots, r_{nn})$, in which x is the probability matrix of the test subsequence and r is a class representative vector,

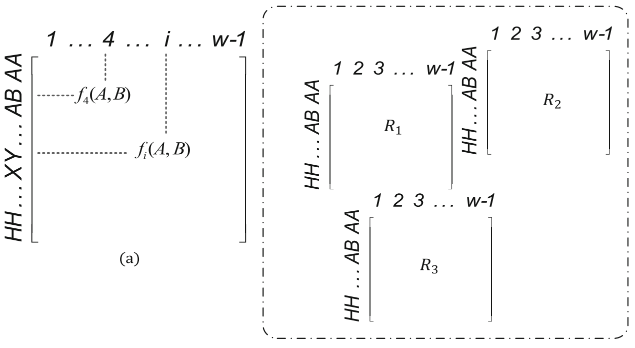


Fig. 3. (a) The probability matrix created from a normal sequence and (b) three representatives of that sequence

$$d = \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{n} \left(x_{ij} - r_{ij}\right)^2 \tag{6}$$

In kNN, the class label with the highest frequency of occurrence among the k-nearest neighbors of the test matrix is assigned to it.

After determining the context of the selected test subsequence as an object, the corresponding probability matrix is loaded and the test data stream is scanned for the presence of anomaly by passing through the detection window of the probabilistic detector (described in Sect. 2). This procedure will continue until an alarm occurs or the end of the test data is reached.

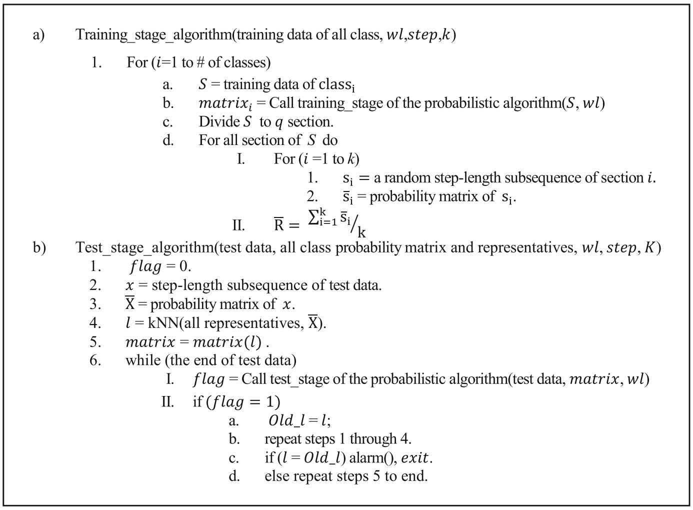
As a result when the training stage ends, each class will have one vector storing the normal data behavior and some representatives, all in the form of the probability matrix.

3.2 The Test Stage

In this stage, a window with a certain size (referred to as *step* in the algorithm) of the test data stream is buffered, and then the probability matrix is created for the captured subsequence. Identifying the context or the class label of the captured subsequence, the kNN classifier is applied to the corresponding matrix.

An alarm makes the context recognition process be repeated for the next *step* size subsequence of the incoming test data stream. If the class label of this subsequence differs from what already was, the alarm is ignored and the class label is changed to the newly recognized one, so its corresponding probability matrix is loaded and scanning the test data stream is continued with this matrix. Moreover, the alarm will be announced, if the new class label is the same as previous one.

This is due to the fact that the probabilistic anomaly detector is bound to a fixed context data stream, so in this case, any changes in the data context raises the alarms that should be ignored as they do not indicate the presence of anomalies, but they are



```
a) Training_stage_algorithm(training data of all class, $wl$,$step$,$k$)
    1. For ($i$=1 to # of classes)
        a. $S$ = training data of $\mathrm{class}_\mathrm{i}$
        b. $matrix_i$ = Call training_stage of the probabilistic algorithm($S$, $wl$)
        c. Divide $S$ to $q$ section.
        d. For all section of $S$ do
            I. For ($i$ =1 to $k$)
                1. $\mathrm{s}_\mathrm{i}$ = a random step-length subsequence of section $i$.
                2. $\bar{\mathrm{s}}_\mathrm{i}$ = probability matrix of $\mathrm{s}_\mathrm{i}$.
            II. $\overline{\mathrm{R}} = \sum_{\mathrm{i}=1}^{\mathrm{k}} \bar{\mathrm{s}}_\mathrm{i} / \mathrm{k}$
b) Test_stage_algorithm(test data, all class probability matrix and representatives, $wl$, $step$, $K$)
    1. $flag$ = 0.
    2. $x$ = step-length subsequence of test data.
    3. $\overline{\mathrm{X}}$ = probability matrix of $x$.
    4. $l$ = kNN(all representatives, $\overline{\mathrm{X}}$).
    5. $matrix = matrix(l)$ .
    6. while (the end of test data)
        I. $flag$ = Call test_stage of the probabilistic algorithm(test data, $matrix$, $wl$)
        II. if ($flag = 1$)
            a. $Old\_l = l$;
            b. repeat steps 1 through 4.
            c. if ($l = Old\_l$) alarm(), $exit$.
            d. else repeat steps 5 to end.
```

Fig. 4. Pseudo code of the context-aware detector, (a) Training stage, (b) Test stage

due to changes in the data context, and with the same context, any alarm should be reported.

Figure 4 depicts a pseudo code of the context-aware detector that the input data stream is assumed to be from some data sources where the anomaly behavior is different.

The first part of this algorithm represents the training stage of the detector which receives the entire classes training data as an input. The input parameter *wl* indicates the detector window length and *step* is the length of the extracted subsequences which are used in creating the representatives and subsequently identifying the data class label. Also, the *k* parameter denotes the number of these subsequences, randomly extracted from one section of a training sequence. The *step* length is suggested to be several times greater than the detector window length, allowing more information to be extracted from one subsequence. Accordingly, the greater the *step* length, the more information that will be stored.

In Fig. 4(a), the same commands are executed on every data class. Initially, the training probability matrix of one class is calculated (step1), and then its training sequence is divided into *q* sections and the class representatives are generated in the form of the probability matrices (step2).

Figure 4(b) represents the test stage of the proposed detector. The input parameter *k* indicates the number of neighbors in the kNN classification rule. Also, the *step* parameter is the length of the buffer used to store the test data, as it is being fed into the detector. Considering that, the *step* size is equal to the previous one in the training stage. But, unlike before, it is desirable to choose a small *step* size to reduce the computational complexity.

Estimating the label of the buffered test subsequence using kNN function and consequently the class of the test data, detection function is performed with the training probability matrix of the specified class $(matrix(l))$. Any event that sets the flag to 1, as long as the class label has not been changed, indicates the presence of anomaly.

4 Constructing the Benchmark Datasets

The process of generating training (normal background) and test data (background plus anomalous events) used for evaluating our detection method are described here.

4.1 Generating the Training and Test Data

This algorithm works by starting with a training-set of purely normal data and then uses the training-set to build a model of normal data in the form of a matrix, so that by referring to this model, the test data will be examined.

First it must be stated that the sensors output data of an embedded system which measures a variable (pressure, temperature, light, humidity, density, speed, road traffic, etc.) in a changing environment can be used for training the proposed detector. For this to work, seven images are taken of a scene with a fixed-position camera in rapid succession. This procedure is repeated for several different light intensities. In other word, our approach is evaluated using a synthetic dataset composed by images taken from a fixed-position camera which differ between each other depending on light intensities. It can be said that the measured variable is a tricolor (RGB) image and the required changing environment is achieved by different light intensities during a day.

Each of the images was cropped to a 50 × 50 pixels image before processing. All image processing were done in MATLAB environment. Any pixel in the image is a three-dimensional array which is a categorical value by itself but the categories are a lot which makes it difficult to evaluate. Converting each array to a decimal number, a numerical vector is returned from an image pixel matrix. The numerical values of each vector were turned into categorical data by matching the values in a given range to a specified character, i.e. the numbers within the range of (0–9) are associated with the symbol “a”, and symbol “b” is assigned to every data value in the (10–19) interval, likewise other values are also categorized. Figure 5(a) depicts the methodology of creating the benchmark dataset. This benchmark dataset also can be used in color image processing applications.

Since, data behavior is also a function of changes in the environment conditions; various lighting levels are used to make the change. As a consequence, collected data for which the behavior is different, affected by the variation in the light intensity, are put separately in distinct classes.

Accordingly, a set of classes that each consist of seven images is obtained. For each class, a normal vector of length 10,000 and about six-hundred test vectors of length 100 are created such that, each element of the normal vector is a symbol with the highest number of occurrence in all seven vectors. Creating the normal vector, the points in other vectors that differ from the norm, are deemed to be anomalies. Also, a test vector belonging to one of the classes is generated by randomly selecting a subset of normal vector as a background data, as well as randomly injecting an anomaly with a specific

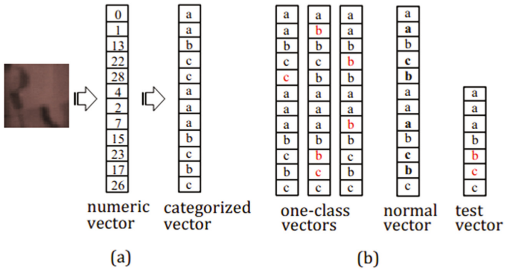


Fig. 5. (a) Constructing categorical data, (b) Generating normal and test vector in a class

length into it. As you can see in Fig. 5(b), a two-length anomaly is injected in the test vector, randomly extracted from the normal data stream, considering that the class members are three vectors derived from an equal situation.

The test data, suitable for evaluating the approach in such a problem, in which data behavior is dependent on environmental conditions, is built by randomly selecting the test vectors from all classes and serializing them consecutively as a contextual test data stream.

5 Experimental Results

The most basic factors for evaluating an anomaly detection system are the false alarm rate (or false-positive rate) and the detection rate (or true-positive rate). It should be noted that the main objective in all anomaly detection methods is to determine whether the test data is anomalous or not, without considering the type and location of the anomaly, so the true-positive rate would be taken into account only in anomalous data sets and false alarms would be evaluated when non-anomalous data sets are fed into the detector. This section describes the experiments performed to evaluate our detection algorithm. Furthermore, the delay and area of the detector is measured using a synthesis tool.

Each of the two detectors, probabilistic and context-aware was written in C++ language and was evaluated on a test dataset containing 460 test vectors; where the anomaly length varies between 1 to 6 symbols. Moreover, four classes are considered, each including a large proportion of training data sets; also the alphabet length in different classes varies between 6 to 9 symbols. The number of representatives has a direct impact on kNN classification accuracy that should be precisely chosen. In this detection algorithm, the same number of representatives was considered for each class.

Table 1 shows the true-positive and false-positive rate of the context-aware anomaly detection algorithm when the detection window size changes between 3 to 5 characters and the number of representatives varies between 4 to 32 vectors. As it can be seen in this table, along with increasing the detection window size, the average rate of true positive and false positive are both increased. This indicates that by increasing the window size, for more test vectors the detector generate alarms that can be due to the reduction in detecting the changes by kNN algorithm. Using this dataset and with a fixed *step* size, increasing the window length reaches the detector to the point that the probability matrices of different classes become more similar and this similarity will be more by taking more representatives which makes the kNN algorithm deteriorates dramatically. As a result, the detector cannot isolate data class changes with anomalies and makes alarms for both of them. As it can be seen in the bottom line of Table 1, the high value of true-positive and false-positive for $w_l = 5$ is because of the weakness of detector adaptation to the changes in data context.

The true-positive and false-positive rate of the Probabilistic method is shown in Table 2. It is worth mentioning that the high rate of false alarm is due to lack of the detector awareness to the context changes. It should also be noted that, the Probabilistic anomaly detector is not capable of differentiating the anomaly from changes in data context and designates these changes as anomalous consequence. Accordingly, the Probabilistic method cannot adapt itself to the new context and will fail in detecting the contextual anomalies. In our experiment, the class of the test data has been changed over time so the high rate of false alarm was generated.

Table 1. True positive and false positive rate of the proposed detector

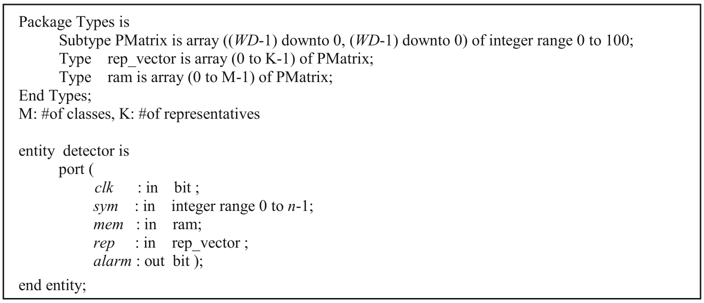
# of representatives	True-positive (%)			False-positive (%)		
	$w_l = 3$	$w_l = 4$	$w_l = 5$	$w_l = 3$	$w_l = 4$	$w_l = 5$
4	67	65	63	2	3	5
8	63	65	69	1	2	2
16	65	63	61	4	7	5
32	63	87	95	5	6	75

Table 2. True positive and false positive rate of the probabilistic detector

Probabilistic method	True-positive (%)	False-positive (%)
	100	100

6 Hardware Implementation

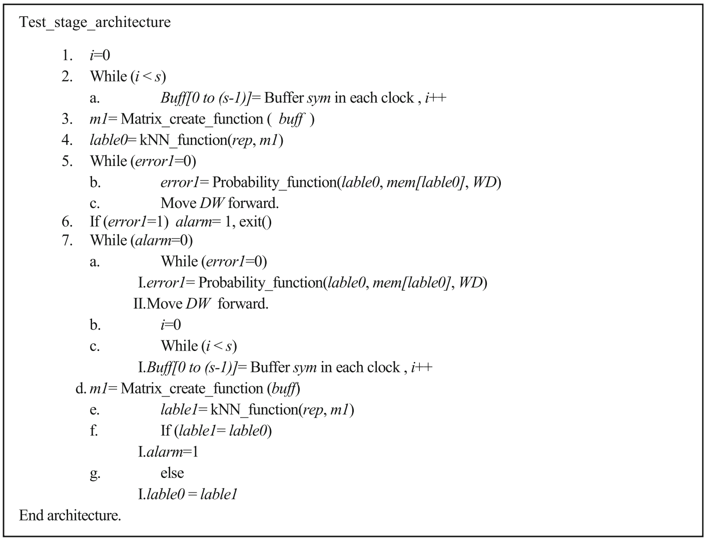
In order to verify the performance of the proposed method, the detector testing stage was implemented in VHDL language and was simulated using Modelsim simulation tool. In our implementation, some ports are defined in order to connect the outputs of the training stage to the detector test circuit. We provide the entity of the design in Fig. 6, in which the parameter n indicates the alphabet length of the training-set and the ports *mem* and *rep* denote all class probability matrices and representatives. The architecture of this entity is presented in Fig. 7. Finally, Leonardo Spectrum synthesis tool is used to transform the hardware description of the proposed method to a gate level netlist corresponding to the 65 nm ASIC technology. Adding that, the proposed detector can be implemented on an ASIC platform included with a microprocessor and a programmable memory integrated with other custom features. This is due to the greater capacity and faster performance of ASIC technology. The area optimization results for different detection window sizes are presented in Table 3.



```
Package Types is
        Subtype PMatrix is array ((WD-1) downto 0, (WD-1) downto 0) of integer range 0 to 100;
        Type    rep_vector is array (0 to K-1) of PMatrix;
        Type    ram is array (0 to M-1) of PMatrix;
End Types;
M: #of classes, K: #of representatives

entity  detector is
        port (
                clk    : in    bit ;
                sym    : in    integer range 0 to n-1;
                mem    : in    ram;
                rep    : in    rep_vector ;
                alarm  : out   bit );
end entity;
```

Fig. 6. Entity declaration of the context-aware detector



```
Test_stage_architecture

    1. i=0
    2. While (i < s)
        a.      Buff[0 to (s-1)]= Buffer sym in each clock , i++
    3. m1= Matrix_create_function ( buff )
    4. lable0= kNN_function(rep, m1)
    5. While (error1=0)
        b.      error1= Probability_function(lable0, mem[lable0], WD)
        c.      Move DW forward.
    6. If (error1=1)  alarm= 1, exit()
    7. While (alarm=0)
        a.      While (error1=0)
                  I.error1= Probability_function(lable0, mem[lable0], WD)
                 II.Move DW  forward.
        b.      i=0
        c.      While (i < s)
                  I.Buff[0 to (s-1)]= Buffer sym in each clock , i++
      d. m1= Matrix_create_function (buff)
        e.      lable1= kNN_function(rep, m1)
        f.      If (lable1= lable0)
                  I.alarm=1
        g.      else
                  I.lable0 = lable1
End architecture.
```

Fig. 7. Pseudo code of the hardware description of the proposed detection algorithm test stage

Table 3. The synthesis results of context-aware detector

w_l	Clock frequency(MHz)	Total area (# of gates)
3	51.0	1804
4	38.2	3769
5	30.7	5793

Table 4. The synthesis results of Probabilistic detector

w_l	Clock frequency(MHz)	Total area (# of gates)
3	84.5	245
4	54.1	572
5	38.9	972

7 Conclusion

In this paper, we propose an anomaly detection method that makes a detection system, aware and capable to react to the changes in its environment. This approach employs the kNN clustering algorithm to identify the context of the test data stream feeding into the detector. After determining the context of a short test subsequence, the corresponding probability matrix is loaded into the detector. The context is assumed to be constant until an alert is generated. Finally, based on the context of the new incoming test subsequence, the generated alarm will be ignored or reported. All detection methods provided in embedded systems are not applicable in changing environments as all of them generate false alarms mainly due to failure in recognizing the context changes. The proposed method can detect an average of 68.83% of anomalies while generating 11.41% of false alarms. To demonstrate the usefulness of the proposed method, several experiments are conducted to demonstrate its effectiveness and efficiency (Table 4).

References

1. Budalakoti, S., Srivastava, A.N., Otey, M.E.: Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **39**, 101–113 (2009)
2. Maxion, R., Tan, K.: Anomaly detection in embedded systems. IEEE Trans. Comput. **51**, 108–120 (2002)
3. Chandola, V.: Anomaly detection for symbolic sequences and time series data, University of Minnesota (2009)
4. Margineantu, D., Bay, S., Chan, P., Lane, T.: Data mining methods for anomaly detection kdd-2005 workshop report. ACM SIGKDD Explor. Newslett. **7**, 132–136 (2005)
5. Hancock, E., Pelillo, M.: Similarity-Based Pattern Recognition. Springer, Heidelberg (2011)
6. Maxion, R., Feather, F.E.: A case study of ethernet anomalies in a distributed computing environment. IEEE Trans. Reliab. **39**, 433–443 (1990)
7. McCluskey, A., Lalkhen, A.G.: Statistics I: data and correlations. Continuing Educ. Anaesth. Crit. Care Pain **7**, 95–99 (2007)
8. Akoglu, L., Tong, H., Vreeken, J., Faloutsos, C.: Fast and reliable anomaly detection in categorical data. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 415–424 (2012)
9. Hayes, M.A., Capretz, M.A.: Contextual anomaly detection framework for big sensor data. J. Big Data **2**, 1–22 (2015)
10. Jiang, Y., Zeng, C., Xu, J., Li, T.: Real time contextual collective anomaly detection over multiple data streams. In: Proceedings of the ODD, pp. 23–30 (2014)
11. Janakiram, D., Adi Mallikarjuna Reddy, V.: Outlier detection in wireless sensor networks using Bayesian belief networks. In: First International Conference on Communication System Software and Middleware, pp. 1–6 (2006)
12. Srivastava, A.N.: Discovering system health anomalies using data mining techniques. In: Proceedings of Joint Army Navy NASA Airforce Conference on Propulsion (2005)

13. Warrender, C., Forrest, S., Pearlmutter, B.: Detecting intrusions using system calls: alternative data models. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy, pp. 133–145 (1999)
14. Zandrahimi, M., Zarandi, H.R., Mottaghi, M.H.: Two effective methods to detect anomalies in embedded systems. Microelectron. J. **43**, 77–87 (2012)

Comparative Analysis of Calculations in Cryptographic Protocols Using a Combination of Different Bases of Finite Fields

Sergey Gashkov[1](✉) and Alexander Frolov[2]

[1] Faculty of Mechanics and Mathematics,
Lomonosov Moscow State University - MSU,
GSP-1, 1 Leninskiye Gory, Main Building, Moscow 119991, Russia
sbgashkov@gmail.com

[2] National Research University Moscow Power Engineering Institute,
Krasnokazarmennaya, 14, Moscow 111250, Russian Federation
abfrolov@gmail.com

Abstract. The chapter introduces a comparative analysis of the complexity of the Tate pairing operation on a supersingular elliptic curve and the complexity of the final exponentiation in the tripartite key agreement cryptographic protocol. The analysis takes into account a possibility of using different bases of finite fields in combination. Operations of multiplication and multiple squaring in the field $GF(2^n)$ and its 4-degree extension, of Tate pairing on supersingular elliptic curve and of final exponentiation are considered separately and in combination. We conclude that the best complexity bound for the pairing and the final exponentiation in the cryptographically significant field $GF(2^{191})$ is provided by the combination of the polynomial basis of this field and 1-type optimal basis of the field expansion.

Keywords: Finite field · Extension of finite field · Optimal normal basis · Combination of bases · Supersingular elliptic curve · Tate pairing · Algorithm with square root extraction · Algorithm without square root extraction · Final exponentiation

1 Introduction

The idea of combining bases to accelerate computations in finite fields was first introduced in [1] on the basis of the estimates of complexity of transformations of bases in the fields possessing the 2- or 3-type optimal normal basis (o.n.b.) [2]. In [3–5], a number of modifications of the multiplication in these bases have been proposed. In particular, in [5] multiplication algorithm in the so-called optimal polynomial basis of type 2 (in the terminology of [1] - almost polynomial basis (a.p.b)) using the multiplication operations in the ring $GF(2)[X]$ is described and estimated. The product is converted into a.p.b. using a permutated o.n.b., i.e. by means of operations without reduction modulo an irreducible polynomial.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_16

Recall that 1-type o.n.b. in the field $GF(2^n)$ occurs if $p = n+1$ is a prime number, and 2 is a primitive root mod p, 2-type o.n.b. or 3-type o.n.b. arise when $p = 2n + 1$ is a prime number, and the field characteristic 2 is a primitive root modulo p. If $p \equiv 3$ (mod 4), and 2 is a quadratic residue, we have 3-type o.n.b, otherwise 2-type o.n.b.

As used in this paper, the field $GF(2^{191})$, has 3-type o.n.b. and its 4-th degree extension has 1-type o.n.b.. Availability o.n.b. allowing to speed up the operation of raising to a power equal to the power of the field characteristics, or the scalar multiplication of the elliptic curve point by the power of the field characteristics, as well as the operation of the square root extracting.

Polynomial basis p.b. of these fields has generators, which are the roots of the irreducible trinomials that simplifies the implementation of the operations of multiplication in these bases. Thus, there is reason to explore the possibility of sharing o.n.b. and p.b. in the implementation of the various stages of cryptographic protocols.

In this paper, we concretize the idea of using combinations of bases in relation to the implementation of the tripartite key agreement protocol [6] in arithmetic supersingular elliptic curve over a cryptographically significant field $GF(2^{191})$: tacking into account security parameter $k = 4$ for supersingular elliptic curve over this field, security of discrete logarithm problem in group of elliptic curve points is equivalent to security of this problem in multiplicative group of order $2^{764} - 1$ [7]. Recall that this protocol is one round. System parameter is a point P of supersingular elliptic curve over the ground field $GF(2^n)$.

Each of the three participants A, B and C selects a private key a, b and c, computes and publishes the public key $KA = aP$, $KB = bP$ and $KC = cP$. Then each of them receives the public keys of other participants, calculates an element $e(bP,cP)$, $e(aP, cP)$ or $e(bP, aP)$ of the field $GF(2^{n \times 4})$ performing the Tate pairing operation e with two points of an elliptic curve and then the operation of the final exponentiation (raising to a power equal to the quotient of the order of the group $GF(2^{n \times 4})^*$ on the order of the elliptic curve). The final step is to calculate the shared secret key by exponentiation of the result to the power a, b and c respectively. The chapter provides upper bounds on the number of elementary operations in the pairing and the final exponentiation phases of the said cryptographic protocol. The rest part of chapter contains Sect. 2 were operation of multiplication and multi squaring in distinct bases of the field $GF(2^{191})$ are considered, Sect. 3 that is devoted to these operations in 4-degree extension of this field comparing their complexity for distinct combinations of bases of basic field and its extension. In Sect. 4 we compare complexity of pairing and final exponentiation operations separately and totally for distinct combinations of bases. In conclusion the comparison results are summarized.

2 Bases and Operations in $GF(2^n)$

Consider a sequence $\beta_i = \alpha^i + \alpha^{-i} \in GF(2^n)$, $n = 191, i \in Z$. The set $\{1, \beta_1, \ldots, \beta_1^i, \ldots, \beta_1^{n-1}\}$ is called a polynomial base (p.b) of $GF(2^n)$, the set $\{\beta_1, \ldots, \beta_1^i, \ldots, \beta_1^n\}$ $(\{\beta_1, \ldots, \beta_1^i, \ldots, \beta_1^{2n}\})$ forms an almost p.b (a.p.b) of $GF(2^n)$ (doubled a.p.b). The set $\{\xi_1, \ldots, \xi_i, \ldots, \xi_n\} = \left\{\beta_1^{2^0}, \ldots, \beta_1^{2^{i-1}}, \ldots, \beta_1^{2^{n-1}}\right\}$ is an optimal normal base (o.n.b) of $GF(2^n)$. The set $\{\beta_1,\ldots,\beta_i,\ldots,\beta_n\}$ (or $\{\beta_1,\ldots,\ \beta_i,\ldots,\beta_{2n}\}$), $\beta_i = \xi_{\pi(i)}, i = 1,\ldots n$, where π is a permutation

$$\pi(i) = \begin{cases} 2^i \bmod p \text{ if } 2^i \bmod p \le n, \\ (p - 2^i) \bmod p \text{ if } 2^i \bmod p > n, \end{cases}$$

$p = 2n + 1$, is called a permuted o.n.b. (p.o.n.b) (or doubled p.o.n.b).

Let $T\,(T^{-1})$ denote the operation of the conversion from a.p.b to p.o.n.b (from p.o.n.b to a.p.b). If $2^k < n < 2^{k+1}$, then the conversion complexity (number of xor-operations) satisfies the recurrent inequality $L_T(n) \le L_T(n - 2^k) + L_T(2^k) + n - 2^k$ with the initial value $L_T\,(2) = 0$. This recurrence can be solved as $L_T(2^k) \le 2^{k-1}(k-2) + 1$ [5]. Note, that the weaker bound $L_T(2^k) \le 2^{k-1}(k) + 1$ was derived in [1] due to the overestimated initial value $L_T(2) = 2$. From this inequality one can obtain estimations $L_T\,(191) = 513$, $L_T(382) = 1227$. Trivially, the complexity of the operation D of the conversion from d. p.o.n.b to p.o.n.b. is n-1.

Following [5], we multiply elements of $GF(2^n)$ represented in a.p.b as elements of the ring $GF(2)[X]$ getting the product in d.a.p.b. Denote this operation $\times_R$. Also following [5] we denote *Bottom*(**a**) and *Top*(**a**) the lower half of product and product after replacing of its lower half with zeros). We implement two multiplication operations in a.p.b:

$\mathbf{y} \times_{apbP} \mathbf{z}$ with result in a.p.b,

$\mathbf{y} \times_{apbN} \mathbf{z}$ with result in p.o.n.b,:

$\mathbf{y} \times_{apbP} \mathbf{z} = Bottom(\mathbf{c}) + T^{-1}\,(D(T(Top(c))))$,

$\mathbf{y} \times_{apbN} \mathbf{z} = T(Bottom(\mathbf{c})) + D(T(Top(c)))$,

where $\mathbf{c} = \mathbf{y} \times_R \mathbf{z}$, «+» is n-bit xor.

It follows that complexity of each of these operations (number of logical operations) $L(\times_{apbP}) = L(\times_{apbN}) = M(n) + L_T(2n) + 2n$ where $M(n)$ is complexity of operation $\times_R$ (transformation D in this case is performed without "xor" - operations).

Implementing $\mathbf{c} = T(\mathbf{y}) \times_R\ T(\mathbf{z})$ instead of $\mathbf{c} = \mathbf{y} \times_R z$ we obtain also two multiplication operations in p.o.n.b:

$\mathbf{y} \times_{ponbP} \mathbf{z}$ with result in a.p.b,

$\mathbf{y} \times_{ponbN} \mathbf{z}$ with result in p.o.n.b.

Complexity of each of these operations $L(\times_{ponbP}) = L(\times_{ponbN}) = M(n) + 2 * L_T\,(n) +\ L_T(2n) + n$.

Denote $\times_{pbN}$ multiplication in p.b. in the field $GF(2^n)$ with minimal polynomial $P_{\text{onb}}(X)$ that root generates o.n.b. It can be performed converting operands to a.p.b., executing $\times_{apbP}$ and converting the product back to p.b. Mentioned converting's are of complexity n. Hence complexity of multiplication $\times_{pbN}$ is $L(\times_{apbP}) + 3n$.

P.b. is organized using the root of trinomial $P_{\text{pb}}(X)$ instead $P_{\text{onb}}(X)$. Let $R(\mathbf{x})$ be the modulo trinomial reduction of complexity $2n$. Then multiplication in p.b. of $GF(2^n)$ is denoted and described as $\mathbf{y} \times_R \mathbf{z} = R(\mathbf{x} \times_R \mathbf{y})$. Its complexity is limited to $M(n)$ + $2n$. Squaring in p.b. is performed by an algorithm that directly take into account the reducing of the vector-result. Below we denote this operation $z_{pb}^{(1)} = \boldsymbol{Z}^2$. Its complexity is limited to n, but symbolically synthesis for n = 191 gave the squaring program with only 99 xor's.

Denote $z_{ponb}^{(j)} = z^{2^j}$ a raising to power 2^j operation implemented to element **z** in p.o.n.b. with result in a.p.b.:

```
for $i = (1, n)$:
$\mathbf{b}[i] = \mathbf{a}[\pi[\pi^{-1}(i) - \mathrm{j}]]$
$\mathbf{c} = T(\mathbf{b})$.
```

Its complexity equals 0, because logical operations absent.

Operation $z_{apb}^{(j)} = T^{-1}(T(z)^{2^j})$ is implemented to element **z** in a.p.b. with result in a.p.b. Its complexity is $2L_T$ (n).

Operation $z_{apbN}^{(j)} = T(z)^{2^j}$ is implemented to element **z** in a.p.b. with result in p.o.n.b. Its complexity is L_T.

Operation $z_{pbN}^{(j)} = T(z)^{2^j}$ is implemented to element **z** in p.b. (for minimal polynomial $P_{\mathrm{onb}}(X)$) with result in p.b. Its complexity is $2L_T(n) + n)$.

Denote $z_{pb}^{(j)}$ a raising to power 2^j operation to element **z** in p. b. (for minimal polynomial $P_{\mathrm{pb}}(X)$) with result in p.b.:

```
$\mathbf{c} = \mathbf{z}$
for $i = (1, j)$:
        $\mathbf{c} = \mathbf{c}_{pb}^{(1)}$
```

Its complexity is bounded by nj (for n = 191 one can take estimate 99j).

In Table 1 there are represented the numbers of logical operations "xor" and "and" (denoted ⊕ and &) and the total numbers of these operations in rows {⊕, &} for multiplication and squaring in distinct bases of $GF(2^{191})$. Here and below we assume implementation the fastest of the stated algorithms for multiplication in the ring $GF(2)[X]$ [8]. Here and below in tables there are represented data derived from estimates of the complexity of operations and confirmed by computer experiment. Column "Δ over the ring" contains numbers of operations without operations X_R of multiplication in the ring.

3 Multiplication and Squaring in $GF(2^{191\times 4})$

The field $GF((2^{191})^4)$ contains a 1-type o.n.b. over the subfield $GF(2^{191})$. Operations of addition, multiplication, and squaring in these bases can be implemented using operations in $GF(2^{191})$ in p.b, a.p.b, or p.o.n.b of this basic field. It follows that there are 6 combinations of bases of basic field and its extension, and each of them can be chosen to implement operations of Tate pairing, final exponentiation, and operation of secret working key computing. Together with operations considered in the second

Table 1. Comparison of Operations in $GF(2^{191})$

Bases of $GF(2^{191})$	Minimal polynomial	Logical operations	Squaring	j-Times squaring	Multiplication	Δ over the ring
a.p.b.	Ponb(X)	⊕	1026	1026	15798	1418
		&	0	0	5724	0
		{⊕,&}	1026	1026	21522	1418
p.o.n.b.	Ponb(X)	⊕	0	0	16311	1931
		&	0	0	5724	0
		{⊕,&}	0	0	22035	1931
p.b	Ponb(X)	⊕	1408	1408	16371	1991
		&	0	0	5724	0
		{⊕,&}	1408	1408	22095	1991
p.b	Ppb(X)	⊕	99	$99j$	14758	378
		&	0	0	5724	0
		{⊕,&}	99	$99j$	20482	378

section, algorithms of these operations use operations of multiplication in 4-degree extension of the field $GF(2^n)$ and operation of raising to power 2^j.

Let $\mathbf{a}^{(j)}_{apb\_4nP}$ denote the operation of raising an element $\mathbf{a}$ to the power 2^j using operation $z^{(j)}_{apb}$ of basic field and p.b. of its extension.

Let $\times_{apb\_4nP}$ be a multiplication using a.p.b. of basic field with multiplication $\times_{apb}$ and p.b. of its extension.

Analogous notation $\mathbf{a}^{(j)}_{apb\_4nN}$, $\mathbf{a}^{(j)}_{ponb\_4nP}$, $\mathbf{a}^{(j)}_{ponb\_4nN}$, $\times_{apb\_4nN}$, $\times_{ponb\_4nP}$, $\times_{ponb\_4nN}$, $\times_{pb\_4nP}$, $\times_{pb\_4nN}$ are used for operations in other combinations of field extension and basic field.

Denote $+_{4n}$ an addition in field extension in any of its basis and any basis of basic field, its complexity equals $4n$.

In can be shown that operations $\times_{apb\_4nN}$, $\times_{ponb\_4nN}$, $\times_{pb\_4nN}$ can be implemented performing 9 multiplications an 22 additions in the field $GF(2^n)$.

So complexity of $\times_{apb\_4nN}$ equals 4 $L(\times_{apbN}) + 22n$. Complexity of operations $\times_{apb\_4nP}$, $\times_{ponb\_4nP}$, $\times_{pb\_4nP}$ exceed these values of $6n$ accordingly numbers of n-bit additions for converting between o.n.b. and p.b. of field extension.

Complexity of multiple squaring is estimated analogously.

Numbers of logical operation for multiplication in distinct bases of the field $GF(2^{191})$ and its extension are presented in Table 2.

Table 2. Comparison of Operations in $GF(2^{191\times 4})$

Base of $GF(2^n)$	Minimal polynomial	Logical and n-bit operations	Numbers of logical operations if there are used the bases of $GF(2^{n\times 4})$ over $GF(2^n)$, $n = 191$:					
			o.n.b.			p.b		
			×	squaring	661-times squaring	×	squaring	661-times squaring
a.p.b	Ponb(X)	⊕	146384	4104	1026	147530	5250	5250
		&	51516	0	0	51516	0	0
		{⊕,&}	197900	4104	1026	199046	5250	5250
p.o.n.b	Ponb(X)	⊕	151001	0	0	152147	1146	1146
		&	51516	0	0	51516	0	0
		{⊕,&}	202517	0	0	203663	1146	1146
p.b	Ppb(X)	⊕	137024	396	65439	138170	1542	261756
		&	51516	0	0	51516	0	0
		{⊕,&}	188540	396	65439	189686	1542	261756

4 Tate Pairing and Final Exponentiation Operations

Let us consider four variants of Tate pairing computing with root extraction on supersingular elliptic curve $Y^2 = X^3 + X + \mathbf{b}$ [9].

(a) A.p.b. of the field $GF(2^{191})$, o.n.b of its extension.

Algorithm $Pairing\_apb\_onb(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{x}, \mathbf{y}, \mathbf{t}_{apb\_onb}, \mathbf{b})$ for pairing of points $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ using pairing parameter $\mathbf{t}_{apb\_onb}$ (an element of the extension field with all coefficients being 0 s and 1 s of the field $GF(2^{191})$), $\mathbf{b} = 1_{apb}$ (identity element represented in a.p.b of $GF(2^n)$).

```
$\mathbf{C}$ = [$1_{apb}$, $1_{apb}$, $1_{apb}$, $1_{apb}$]
$\mathbf{t} = \mathbf{t}_{apb\_onb}$
$\mathbf{s} = \mathbf{t}^{(1)}_{apbP\_4nN}$
for $i$ = (1,$n$):
    $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{apbP}{}^{(1)}$
    $\boldsymbol{\beta} = \boldsymbol{\alpha}_{apbP}{}^{(1)}$
    $\mathbf{z} = \boldsymbol{\alpha}+\mathbf{x}$
    $\mathbf{v} = \boldsymbol{\alpha}\times_{apbP}\mathbf{x}$
    $\mathbf{w} = \mathbf{z} + \mathbf{v}+\boldsymbol{\beta}+\mathbf{y}+1_{apb}$
    $\mathbf{u}$ = [$\mathbf{z}\times_{apbP}\mathbf{t}$[0]), $\mathbf{z}\times_{apbP}\mathbf{t}$[1]), $\mathbf{z}\times_{apbP}\mathbf{t}$[2]), $\mathbf{z}\times_{apbP}\mathbf{t}$[3]),),
    $\mathbf{v} = \mathbf{z}+1_{apb}$
    $\mathbf{r}$ = [$\mathbf{v}\times_{apbP}\mathbf{t}_{apb}\mathbf{s}$ [0], $\mathbf{v}\times_{apbP}\mathbf{s}$[1], $\mathbf{v}\times_{apbP}\mathbf{s}$[2], $\mathbf{v}\times_{apbP}\mathbf{s}$[3]]
    $\mathbf{v}$ = [$\mathbf{w},\mathbf{w},\mathbf{w},\mathbf{w}$] + $_{4n}\mathbf{u}$ + $_{4n}\mathbf{r}$
    $\mathbf{C} = \mathbf{C}\times_{apbP\_4nN}\boldsymbol{v}$
    $\mathbf{x} = \mathbf{x}^{(n-1)}_{apb}$
    $\mathbf{y} = \mathbf{y}^{(n-1)}_{apb}$
```

Complexity of this algorithm estimated accordingly numbers if multiplication, addition and squaring operations in them:

$$\mathrm{L}_{Pairing\_apb\_onb}(n) = 191(2L(\mathbf{z}^{(1)}_{apb}) + 2\mathrm{L}(\mathbf{z}^{(190)}_{apb}) + L(\times_{apbP}) + L(\times_{apb\_4nN}) + 15L(+)).$$

Remark that here and below multiplication with multiples **t** or **s** containing trivial elements are not taken into account in assessing complexity of pairing, $L(+)$ is complexity of addition in $GF(2^{191})$.

(b) P.o.n.b. of the field $GF(2^{191})$, o.n.b of its extension.

Algorithm $Pairing\_ponb\_onb(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{x}, \mathbf{y}, \mathbf{t}_{ponb\_onb}, \mathbf{b})$ for pairing of points $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ using pairing parameter $\mathbf{t}_{ponb\_onb}$ when $\mathbf{b} = 1_{ponb}$ (that is, the identity element represented in p.o.n.b. of $GF(2^n)$) differs from just considered only in the type used in operations in the notation of which "apb" is replaced by "ponb", 1_{apb} is replaced by 1_{ponb}.

Hence complexity of this pairing operation is represented by formula

$$L_{Pairing\_ponb\_onb}(n) = 191(2L(\mathbf{z}^{(1)}_{ponb}) + 2L\mathbf{z}^{(190)}_{ponb} + L(\times_{ponbP}) + L(\times_{ponb\_4nN}) + 15L(+)).$$

(c) A.p.b. of the field $GF(2^{191})$, and p.b. of its extension.

Algorithm $Pairing\_apb\_pb(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{x}, \mathbf{y}, \mathbf{t}_{apb\_pb}, \mathbf{b})$ for pairing of points $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ using pairing parameter $\mathbf{t}_{apb\_pb}$ when $\mathbf{b} = 1_{apb}$.

```
$\mathbf{C} = [1_{apb}, 0_{apb}, 0_{apb}, 0_{apb}]$
$\mathbf{t} = \mathbf{t}_{apbpb}$
$\mathbf{s} = \mathbf{t}_{apbpbapbP\_4nPb}{}^{(1)}$
for $i = (1,n)$:
    $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{apbP}{}^{(1)}$
    $\boldsymbol{\beta} = \boldsymbol{\beta}_{apbP}{}^{(1)}$
    $\mathbf{z} = \boldsymbol{\alpha}+\mathbf{x}$
    $\mathbf{v} = \boldsymbol{\alpha}\times_{apbP}\mathbf{x}$
    $\mathbf{w} = \mathbf{z} + \mathbf{v} + \boldsymbol{\beta} + \mathbf{y} + 1_{apb}$
    $\mathbf{u} = [\mathbf{z}\times_{apbP}\mathbf{t}[0], \mathbf{z}\times_{apbP}\mathbf{t}[1], \mathbf{z}\times_{apbP}\mathbf{t}[2], \mathbf{z}\times_{apbP}\mathbf{t}[3]]$
    $\mathbf{v} = \mathbf{z}+1_{apb}$
    $\mathbf{r} = [\mathbf{v}\times_{apbP}\mathbf{s}[0],\mathbf{v}\times_{apbP}\mathbf{s}[1],\mathbf{v}\times_{apbP}\mathbf{s}[2], \mathbf{v}\times_{apbP}\mathbf{s}[3]]$
    $\mathbf{v} = [\mathbf{w}, 0_{apb}, 0_{apb}, 0_{apb}] + {}_{4n}\mathbf{u} + {}_{4n}\mathbf{r}$
    $\mathbf{C} = \mathbf{C}\times_{apbP\_4nPb}\mathbf{v}$
    $\mathbf{x} = \mathbf{x}^{(n-1)}_{apbP}$
    $\mathbf{y} = \mathbf{y}^{(n-1)}_{apbP}$
```

Its complexity is the following:

$$\begin{aligned} L_{Pairing\_apb\_pb}(n) = 191(&2L(\mathbf{z}^{(1)}_{apb}) + 2L(\mathbf{z}^{(190)}_{apb}) \\ &+ L(\times_{apbP}) + \mathrm{L}(\times_{apb\_4nP}) + 11\mathrm{L}(+)). \end{aligned}$$

(d) P.o.n.b. of the field $GF(2^{191})$, p.b. of its extension.

Algorithm $Pairing\_ponb\_pb(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{x}, \mathbf{y}, \mathbf{t}_{ponb\_pb}, \mathbf{b})$ for pairing of points $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ using pairing parameter $\mathbf{t}_{ponb\_pb}$ when $\mathbf{b} = 1_{ponb\_pb}$ can be obtained from the considered algorithm $Pairing\_apb\_pb(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{x}, \mathbf{y}, \mathbf{t}_{apb\_pb}, \mathbf{b})$ via substitution of indices of operations, pairing parameter, and the field identity element. Its complexity is estimated by formula

$$\mathrm{L}_{Pairing\_ponb\_pb}(n) = 191(2L(\mathbf{z}^{(1)}_{ponb}) + 2L(\mathbf{z}^{(190)}_{aonb}) \\ + L(\times_{ponbP}) + L(\times_{ponb\_4nP}) + 11L(+)).$$

Now consider two variants of Tate pairing computing without root extraction on supersingular elliptic curve $Y^2 = X^3 + X + \mathbf{b}$ [9].

(a) P.b. of the field $GF(2^{191})$, o.n.b. of its extension.

Algorithm $Pairing\_pb\_onb(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{x}, \mathbf{y}, \mathbf{t}_{pb\_onb}, \mathbf{b})$ for pairing of points $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ using pairing parameter $\mathbf{t}_{pb\_onb}$ when $\mathbf{b} = 1_{pb}$.

```
$\mathbf{C} = [1_{pb}, 1_{pb}, 1_{pb}, 1_{pb}]$
$\mathbf{t} = \mathbf{t}_{pbonb}$
$\mathbf{s} = \mathbf{t}_{pb\_4nN}{}^{(1)}$
$\mathbf{u} = \mathbf{x}^{(1)}_{\mathrm{pb}}$
$\mathbf{v} = \mathbf{u}$
$\mathbf{y} = \mathbf{y}^{(1)}_{\mathrm{pb}}$
for $i = (1,n)$:
    $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(4)}_{\mathrm{pb}}$
    $\boldsymbol{\beta} = \boldsymbol{\beta}^{(4)}_{\mathrm{pb}}$
    $\mathbf{w} = \boldsymbol{\alpha}\times_{\mathrm{bp}}(\mathbf{v} + 1_{\mathrm{pb}}) + \mathbf{u}+\mathbf{y} + \mathbf{b}+((n\text{-}1)/2)_{\mathrm{pb}}$
    $\mathbf{v} = \boldsymbol{\alpha}+\mathbf{v}$
    $\mathbf{r} = \mathbf{v}+1_{pb}$
    $\mathbf{a} = [\mathbf{w} + \mathbf{v}\times_{pb}\mathbf{t}\,[0] + \mathbf{r}\times_{pb}\mathbf{s}[1], \mathbf{w} + \mathbf{v}\times_{pb}\mathbf{t}_{pb}[2] + \mathbf{r}\times_{pb}\mathbf{s}[3],$
    $\mathbf{w} + \mathbf{v}\times_{pb}\mathbf{s}\,[0] + \mathbf{r}\times_{pb}\mathbf{s}[1], \mathbf{w} + \mathbf{v}\times_{pb}\mathbf{s}[2] + \mathbf{r}\times_{pb}\mathbf{s}[3]]$
    $\mathbf{C} = \mathbf{C}^{(1)}_{\mathrm{pb}\_4nN}\times_{pb\_4nN}\mathbf{a}$
    $\mathbf{u} = \mathbf{u} + \mathbf{v}+1_{pb}$
    $\mathbf{v} = \mathbf{v} + 1_{pb}$
```

Its complexity is estimated as follows:

$$L_{Pairing\_pb\_onb}(n) = 191(2L(\mathbf{z}^{(1)}_{pb}) + 2\mathrm{L}(\mathbf{z}^{(4)}_{pb}) + 2L(\mathbf{z}^{(190)}_{aonb}) \\ + L(\times_{pb}) + 2\mathrm{L}(\times_{\mathrm{pb}\_4nN}) + \mathrm{L}\left(\mathbf{a}^{(1)}_{ab\_4nN}\right) + 16\mathrm{L}(+)).$$

(b) P.b. of the field $GF(2^{191})$, p.b. of its extension.

Algorithm $Pairing\_pb\_pb(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{x}, \mathbf{y}, \mathbf{t}_{pb\_pb}, \mathbf{b})$ for pairing of points $P = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $Q = (\mathbf{x}, \mathbf{y})$ using pairing parameter $\mathbf{t}_{pb}$ when $\mathbf{b} = 1_{pb}$ differs from just considered in four rows:

$\boldsymbol{C} = [1_{pb}, 0_{pb}, 0_{pb}, 0_{pb}]$
$\mathbf{t} = \mathbf{t}_{pb\_pb}$
$\mathbf{s} = \mathbf{t}^{(1)}_{pb\_4nP}$
$\mathbf{C} = \mathbf{C}^{(1)}_{\mathrm{pb}\_4nP} \times_{\ pb\_4nP}$

Its complexity is represented by formula

$$\mathrm{L}_{Pairing\_pb\_onb}(n) = 191(2L(z^{(1)}_{pb}) + 2L(z^{(4)}_{pb}) + 2L(z^{(190)}_{aonb}) \\ + L(\times_{pb}) + 2L(\times_{\mathrm{pb}\_4nP}) + L\left(\mathbf{a}^{(1)}_{ab\_4nP}\right) + 16L(+)).$$

Table 3 presents data on the number of logical operations executed considered pairing algorithms on supersingular elliptic curve $Y^2 = X^3 + X + 1$ over $GF(2^{191})$ (1 corresponds to 29910607 "xor", or 10875600 "and" or 43094757 of both these operations). In the tables below we also provide better bounds (given in parentheses) obtained via conversion to a basis with faster implementation of the corresponding operation.

Table 3. Comparison of complexity of pairing algorithms

Base of $GF(2^n)$	Minimal polynomial	Logical and n-bit operations	Relative numbers of logical operations if there are used the bases of $GF(2^{n \times 4})$ over $GF(2^n)$, $n = 191$	
			o.n.b.	p.b
			Algorithms with root extraction	
a.p.b	Ponb(X)	⊕	≈1.0753	≈1.0826 (≈1.0753)
		&	≈1.0053	≈1.0053
		{⊕,&}	≈1.0551	≈1.0605 (≈1.0551)
p.o.n.b	Ponb(X)	⊕	≈1.0826 (≈1.0753)	≈1.0928 (≈1.0754)
		&	≈1.0053	≈1.0053
		{⊕,&}	≈1.0590 (≈1.0551)	≈1.0680 (≈1.0551)
			Algorithms without root extraction	
p.b	Ppb(X)	⊕	1	≈1.0122(1)
		&	1	≈1
		{⊕,&}	1	≈1.0089(1)

For the considered supersingular elliptic curve over the field $GF(2^{191})$, the final exponent is

$d = 30916300184138066757562815128236335891970416695496879296716024089598401293785795944029375276012993493222266694949077877984987359180793018784436808613949303377539749281529855.$

Taking into account that in binary expansion of this number the units take the places 0–95, 97–190, 192–381, 478, and 573 one can represent this exponent as

$$\begin{aligned} d &= (((2^{95}+1)2^{286} + \left(2^{190}-1\right))2^{95} + \left(2^{94}-1\right))2^{97} + \left(2^{96}-1\right) \\ &= \left(\left(a_0 2^{286} + a_1\right)2^{95} + a_2\right)2^{97} + a_3. \end{aligned}$$

As a corollary, final exponentiation algorithm corresponds to the formula $\mathbf{x}^d = (((\mathbf{y}_0)^{2^{286}}\mathbf{y}_1)^{2^{95}}\mathbf{y}_2)^{2^{97}}\mathbf{y}_3$, where $\mathbf{y}_0 = \mathbf{x}^{a_0} = \mathbf{x}^{95}\mathbf{x}$ and the remaining elements $\mathbf{y}_1 = \mathbf{x}^{a_1}, \mathbf{y}_2 = \mathbf{x}^{a_2}, \mathbf{y}_3 = \mathbf{x}^{a_3}$, can be computed by the additive chain 1,2,4,8,10,14,20,40,80,94,96,160,180,190 of lengths 13.

This allows obtaining the following program of final exponentiation using a.p.b. of basic field and p.o.n.b. of its extension.

$x = a$;	$\mathbf{v} = \mathbf{x}^{(1)}_{apbP\_4N}$;	$\mathbf{z}_1 = \mathbf{v}\times_{apbN\_4N}\mathbf{x}$;	$\mathbf{z} = \mathbf{v}\times_{apbN\_4N}\mathbf{z}_1$;
$v = z^{(2)}_{1ponbP\_4N}$;	$\mathbf{v} = \mathbf{z}^{(4)}_{ponbP\_4N}$;	$\mathbf{z} = \mathbf{y}\times_{apbN\_4N}\mathbf{z}$;	$\mathbf{v}_2 = \mathbf{x}^{(2)}_{apbP\_4N}$;
$\mathbf{v} = \mathbf{z}\times_{apbP\_4nN}\mathbf{v}$;	$\mathbf{z}_2 = \mathbf{v}\times_{apbP\_4nN}\mathbf{z}_1$;	$\mathbf{v} = \mathbf{x}^{(4)}_{apbP\_4N}$;	$\mathbf{z} = \mathbf{v}\times_{apbP\_4nN}\mathbf{z}_2$;
$\mathbf{v}_{10} = \mathbf{x}\ ^{(10)}_{apbP\_4N}$;	$\mathbf{z}_3 = \mathbf{z}_2\times_{apbP\_4nN}\mathbf{z}_2$;	$\mathbf{z}_4 = \mathbf{z}_2\times\ _{apbN\_4nN}\mathbf{z}_2$;	$\mathbf{v}_3 = \mathbf{x}\ ^{(20)}_{apbP\_4N}$;
$\mathbf{z} = \mathbf{v}_3\times_{apbP\_4nN}\mathbf{z}_3$;	$\mathbf{z}_5 = \mathbf{v}\times_{apbN\_4nN}\mathbf{z}_4$;	$\mathbf{x}\ ^{(40)}_{apbP\_4N}$;	$\mathbf{z} = \mathbf{v}\times_{apbP\_4nN}\mathbf{z}_5$;
$\mathbf{z}_6 = \mathbf{z}_5\times\ _{apbP\_4nN}\mathbf{z}$;	$\mathbf{v} = \mathbf{x}\ ^{(14)}_{apbP\_4N}$;	$\mathbf{z} = \mathbf{v}\times\ _{apbP\_4nN}\mathbf{z}$;	$\mathbf{y}_2 = \mathbf{v}_1\times\ _{apbP\_4nN}\mathbf{z}$;
$\mathbf{z} = \mathbf{v}_2\times_{apbP\_4nN}\mathbf{y}_2$;	$\mathbf{y}_3 = \mathbf{z}_1\times\ _{apbP\_4nN}\mathbf{z}$;	$\mathbf{v} = \mathbf{x}\ ^{(80)}_{apbP\_4N}$;	$\mathbf{z} = \mathbf{z}_6\times\mathbf{v}$;
$\mathbf{z} = \mathbf{v}_6\times\ _{apbP\_4nN}\mathbf{z}$;	$\mathbf{z} = \mathbf{v}_3\times\ _{apbP\_4nN}\mathbf{z}$;	$\mathbf{z} = \mathbf{v}_3\times\ _{apbP\_4nN}\mathbf{z}$;	$\mathbf{z} = \mathbf{v}_3\times\ _{apbP\_4nN}\mathbf{z}_4$;
$\mathbf{z} = \mathbf{v}_{10}\times\ _{apbP\_4nN}\mathbf{z}$;	$\mathbf{y}_1 = \mathbf{z}_2\times\ _{apbP\_4nN}\mathbf{z}$;	$\mathbf{z} = \mathbf{x}^{(95)}_{ponbP\_4N}$;	$\mathbf{y}_0 = \mathbf{z}\times\ _{apbP\_4nN}\mathbf{x}$;
$\mathbf{z} = \mathbf{y}^{(286)}_{0\ ponbP\_4N}$;	$\mathbf{z} = \mathbf{z}\times_{apbN\_4nN}\mathbf{y}_1$;	$\mathbf{z} = z^{(95)}_{ponbP\_4N}$;	$\mathbf{z} = \mathbf{z}\times_{apbP\_4nN}\mathbf{y}_2$;
$\mathbf{z} = \mathbf{z}\ ^{(97)}_{apbP\_4N}$;	$\mathbf{z} = \mathbf{z}\times_{apbP\_4nN}\mathbf{y}_3$;	$c = z.$	

Programs for other combination of fields bases differ only by operation notation. These programs contain 17 multiplication and 14 multi squaring's in the field $GF(2^{191\times 4})$. It is easy to write formula of complexity of these operations and compute their values that are given in Table 4 (1 corresponds to 3421756 logical operations "xor" and "and"). In each case 374 additions, 153 multiplications and 2644 squaring's in $GF(2^{191})$ are executed.

Table 4. Final exponentiation, n = 191

Base of $GF(2^n)$	Minimal polynomial	Logical and n-bit operations	Bases of the field $GF(2^{n\times 4})$ over $GF(2^n)$, $n = 191$:	
			o.n.b.	p.b.
a.p.b.	Ponb	{⊕,&}	1	≈1.0103(≈1.0004)
p.o.n.b.	Ponb	{⊕,&}	≈1.0061(≈1.003)	≈1.0165 (≈1.0005)
p.b.	Ppb	{⊕,&}	≈1.0134	≈1.0192(≈1.0135)

In three partite key agreement protocol, final exponentiation is performed after pairing operation. In Table 5 there are represented total numbers of logical operations for implementations of this composition in distinct combinations of bases (1 corresponds to 44310956 logical operations “xor” and “and”).

Table 5. Comparison of composition of pairing and final exponentiation, n = 191

Bases of $GF(2^n)$	Minimal polynomial	Bases of the field $GF(2^{n\times 4})$ over $GF(2^n)$, $n = 191$:	
		o.n.b.	p.b.
a.p.b	Ponb(X)	≈1.0498	≈1.0555(≈1.0498)
p.o.n.b	Ponb(X)	≈1.05381(≈1.0498)	≈1.0629(≈1.0499)
		Algorithms without root extraction and final exponentiation	
p.b	Ppb(X)	1	≈1.0087(≈1)

5 Conclusion

In this chapter, implementation of algebraic operations in finite fields possessing 2-type or 3- type optimal normal basis and in its 4-degree extension has been comparatively considered taking into account possibility of using distinct combination of bases. Comparative data were also obtained on the complexity of the implementation of pairing and final exponentiation operations in a three-partite key agreement protocol. Based on these data, we can conclude that although for final exponentiation the best is combination of almost polynomial basic of the base field and optimal normal basis of its extension, pairing and final exponentiation are performed faster in polynomial basis of $GF(2^{191})$ and optimal normal basis of its extension. At the same time, it can be noted that the differences in the complexity of implementation with the use of different combinations of bases are not so significant. The advantage of a polynomial basis of the base field is a consequence of the peculiarities of the pairing algorithm without root extraction.

Acknowledgements. This research was supported by the Russian Foundation for Basic Research, project 17-01-00485a. The authors are grateful to Igor Sergeev for editing and anonymous reviewers for comments.

References

1. Bolotov, A.A., Gashkov, S.B.: On quick multiplication in normal bases of finite fields. Discrete Math. Appl. **11**(4), 327–356 (2001)
2. Mullin, R.C., Onyszchuk, I.M., Vanstone, S.A., Wilson, R.M.: Optimal normal bases in *GF*(p^n). Discrete Appl. Math. **22**, 149–161 (1988/1989)

3. Shokrollahi, J.: Efficient implementation of elliptic curve cryptography on FPGA. PhD thesis, Universität Bonn (2007)
4. von zur Gathen, J., Shokrollahi, A., Shokrollahi, J.: Efficient multiplication using type 2 optimal normal bases. In: WAIFI 2007. LNCS, pp. 55–68 (2007)
5. Bernstein, D.J., Lange, T.: Type-II optimal polynomial bases. In: Arithmetic Finite Fields, Proceedings. LNCS, vol. 6087, pp. 41–61 (2010)
6. Joux, A.: A one round protocol for tripartite Diffie-Hellman. In: ANTS 2000. LNCS, vol. 1838, pp. 385–394 (2000)
7. Menezes, A.J., Vanstone, S., Okamoto, T.: Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Trans. Inform. Th. **IT-39**, 1639–1646 (1993)
8. Bernstein, D.J.: Minimum number of bit operations for multiplication. http://binary.cr.yp.to/m.html, (Accessed 2009)
9. Kwon, S.: Efficient tate pairing computation for supersingular elliptic curves over binary fields. Cryptology ePrint archive, Report 2004/303 (2004)

Dynamic Redundancy in Communication Network of Air Traffic Management System

Igor Kabashkin(✉)

Transport and Telecommunication Institute, Lomonosova 1, Riga 1019, Latvia
kiv@tsi.lv

Abstract. In the paper the channel reliability of communication network with common set of standby radio stations for real conditions of Air Traffic Management (ATM) is discussed. ATM systems represent essential infrastructure that is critical for flight safety and communication is a key element in the present ATM system. As additional method of improving the reliability of the ATM communication network a dynamic management of backup modes with two-stage mode of redundant elements is proposed. Mathematical model of the channel reliability is developed. Comparative analysis of redundancy effectiveness for developed and used structure of ATM communication network is performed.

Keywords: Reliability · Redundancy · Air traffic management · Controller · Communication network

1 Introduction

Air Traffic Management (ATM) systems represent essential infrastructure that is critical for flight safety. Communication is a key element in the present ATM system. Communication between air traffic controllers and pilots remains a vital part of air traffic control operations, and communication problems can result in hazardous situations. Analysis of aviation accidents has identified that a breakdown in effective human communication has been a causal or contributing factor in the majority of accidents [1].

There are different types of air traffic controllers:

- Tower controllers direct the movement of vehicles on runways and taxiways. Most work from control towers, watching the traffic they control.
- Approach and departure controllers ensure that aircraft traveling within an airport's airspace maintain minimum separation for safety.
- En route controllers monitor aircraft once they leave an airport's airspace.

The modern ATM system has independent direct communication channels (CC) for each controllers operating at different radio frequencies f_i, $i = \overline{1, m}$, where m is number of CC. The amount of the CC is determined by the structure of ATM in the area of a specific airport and provides independent interaction with the aircrafts for all controllers. Technical support of controller-pilot communication carried out by means of radio stations (RS). Interoperability of technical means and controllers in ATM communication network is provided by voice communications system which is a

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_17

state-of-art solution for air traffic control communication. The modern approach to system design focuses on providing high-availability solutions that are based on reliable equipment and on redundancy strategies tailored to customers' needs and requirements.

Currently, the main method of improving the reliability of controller's CC is to duplicate equipment to provide communications on each frequency of interaction ground-to-air channel (Fig. 1). Each of the *m* controllers communicates with aircraft using the main radio station (RS) as basic hardware. After the failure of main RS, it switches into a work with the backup (redundant radio station).

Unfortunately the economic efficiency of used fault-tolerance approach is low. In the paper another redundant method for communication network of ATM system is discussed.

The rest of this paper is organized as follows. In Sect. 2 some important works in the area of reliability with redundancy are reviewed. In Sect. 3 the main definitions and assumptions are presented and models of ATM communication network reliability with common set of standby radio stations for reservation of different ATM communication channels are proposed. In Sect. 4 the conclusions are presented.

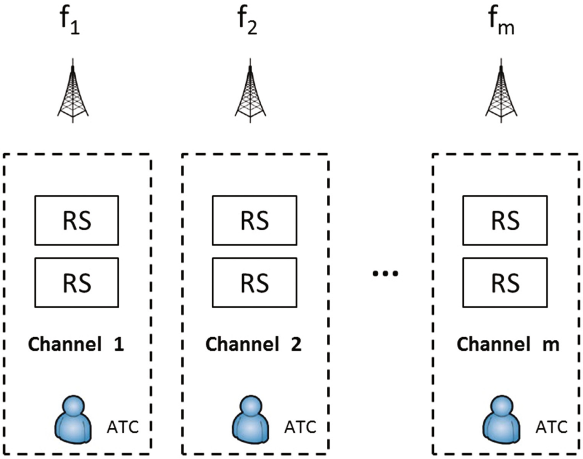


Fig. 1. Redundant communication network of ATM system

2 Related Works

The reliability of duplicated channels is well studied in the literature [2, 3].

One of the methods to increase efficiency of redundancy in the structures with identical elements is the allocation of the common group of reserve elements. The *k-out-of-n* system structure is a very popular type of redundancy in fault tolerant systems. The term *k-out-of-n* system is used to indicate an *n*-component system that works (or is "good") if and only if at least *k* of the *n* components work (or are good). This system is called a *k-out-of-n:G* system. The works [4–6] provide improved versions of the method for reliability evaluation of the *k-out-of-n:G* system. Liu [7] provides an analysis of the *k-out-of-n:G* system with components whose lifetime distributions are not necessarily exponential.

An n-component system that fails if and only if at least k of the n components fail is called a *k-out-of-n:F* system [8]. The term *k-out-of-n* system is often used to indicate either a G system or an F system or both. The *k-out-of-n* system structure is a very popular type of redundancy in fault-tolerant systems. It finds wide applications in telecommunication systems [9, 10]. This model can be used for analyse of reliability of ATM communication network with k controllers and n RS provided availability of CC.

In real conditions it is important to know not the reliability of communication network at whole but each selected CC for controller. The channel reliability problem in standby system consisting of independent elements with some units used as a universal component standby pool is formulated in [11]. In [12] the reliability of selected communication channels with common set of standby radio station in the system with periodical sessions of communications for real conditions of ATM is discussed and advanced model of the channel reliability is developed. In [13] model of the channel reliability in the ATM communication network based on embedded cloud technology in the real conditions of operation is developed.

In this paper we investigate the reliability of selected communication channel with common set of redundant radio stations and with dynamic management of backup modes in ATM communication network.

3 Model Formulation and Solution

The following symbols have been used to develop equations for the models:

λ	– Failure Rate for RS
μ	– Repair Rate for RS
m	– Number of communication channels
n	– Number of RS in common set of redundant radio stations
A_1	– Channel Availability with active backup mode of redundant elements
A_2	– Channel Availability with dynamic management of two-stage backup modes
l	– Number of repair bodies

In this paper we investigate a repairable redundant communication network of ATM system with $N = m + n$ radio stations, m of which are main RS and n radio stations are used as a universal component standby pool (Fig. 2).

As an additional method of improving the reliability of the system we will use a dynamic management of backup modes with two-stage mode of reserve elements.

Let the two-stage group reserve contains n elements, of which $r < n$ are at the first stage of readiness, and $n - r$ are located on the second stage of readiness. At the same time we assume that the second group of redundant elements are in cold standby (standby components do not fail when they are in standby $\lambda_2 = 0$) and can fail only after their transfer to the first stage of the redundancy. Redundant elements of the first group are in hot standby (standby components have the same failure rates as the main components $\lambda_1 = \lambda$). In case of failure of main operating radio station it without delay is replaced by the backup station from the first stage of readiness. At the same time the redundant element of the second stage of readiness is transferred to the group with first stage of readiness.

The above two-stage model of redundancy required to support readiness on the first stage only for z elements, which provide reliable level of operation. In [14] it is shown that if $\lambda_1 > \lambda_2$ at the first stage is optimal to have only one no-fault element. For our case this condition is satisfied, so we can accept that $z = 1$.

For system with $1 \leq l \leq n$ number of repair bodies the behaviour of the examined system is described by the Markov Chain state transition diagram (Fig. 3), where: H_i – state with i failed RS, but in the selected channel there is a workable RS; H_{iI} – state with i + 1 failed RS, in the selected channel there is no a workable RS.

On the base of this diagram the system of Chapman–Kolmogorov's equations can be writing in accordance with the general rules [15].

By solving the resulting system of equations, we can obtain an expression for availability of selected communication channel:

$$A = 1 - \sum_{\forall i,j} P_{ijI} = \frac{a_1 + a_2}{a_1 + a_2 + a_3}, \tag{1}$$

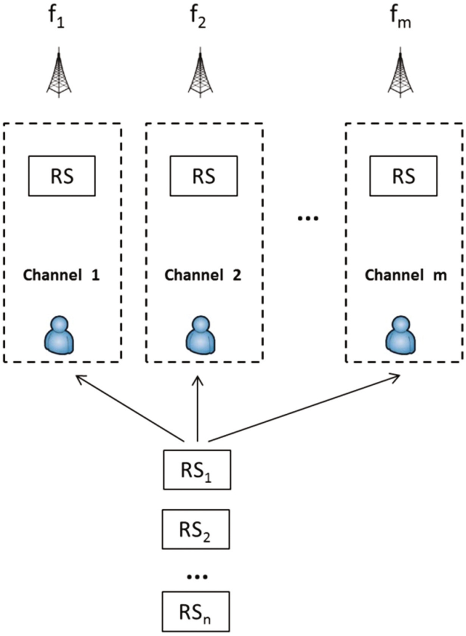


Fig. 2. Communication network of ATM system with common set of standby elements

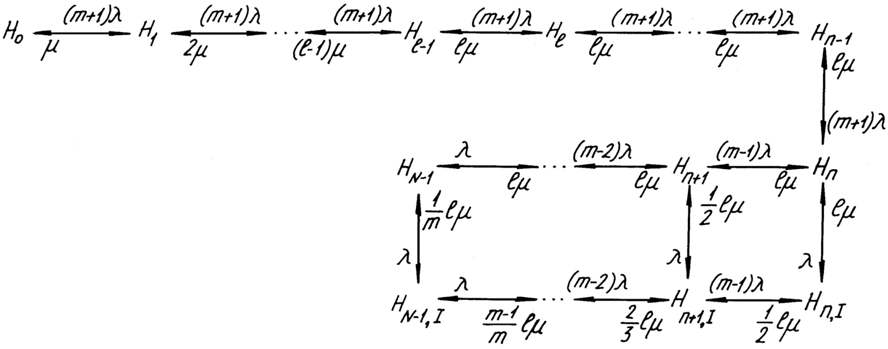


Fig. 3. Markov chain state transition diagram $(1 \leq l \leq n)$

where

$$
\begin{aligned}
a_1 &= \sum_{i=0}^{l} \frac{(m+1)^i}{i!}\gamma^i + \frac{l^l}{l!}\sum_{i=l+1}^{n-1} (m+1)^i \omega^i,\\
a_2 &= \frac{(m+1)^n l^l}{l!}\sum_{i=0}^{m-1} i! \binom{i}{m-1} \omega^{n+i},\\
a_3 &= \frac{(m+1)^n l^l}{l!}\sum_{i=0}^{m-1} (i+1)! \binom{i}{m-1} \omega^{n+i+1},\\
\gamma &= \frac{\lambda}{\mu},\ \omega = \frac{\gamma}{l}.
\end{aligned}
$$

For the system with $n \leq l \leq N$ number of repair bodies the Markov Chain state transition diagram of the system is shown at the Fig. 4. On the base of this diagram the system of Chapman–Kolmogorov's equations can be writing in accordance with the general rules [14].

By solving the resulting system of equations, we can obtain an expression for availability of selected communication channel in accordance of (1), where

$$
\begin{aligned}
a_1 &= \sum_{i=0}^{n-1} \frac{(m+1)^i}{i!}\gamma^i,\\
a_2 &= (m+1)^n \left\{ \frac{\gamma^n}{(n+1)!} \left[n+1+\sum_{i=1}^{l-n} i! \binom{i}{m-1} \gamma^i \right] + \frac{(m-1)! l^l}{l!} \sum_{i=1}^{N-l-1} \frac{\omega^{l+i}}{(N-l-i-1)!} \right\},\\
a_3 &= (m+1)^n \left[\frac{1}{n!} \sum_{i=0}^{l-n} \binom{i}{m-1} \binom{n}{n+i+1}^{-1} \gamma^{n+i+1} + \frac{(m-1)! l^l}{l!} \sum_{i=1}^{N-l-1} \frac{l-n+i+1}{(N-l-i-1)!} \omega^{l+i+1} \right].
\end{aligned}
$$

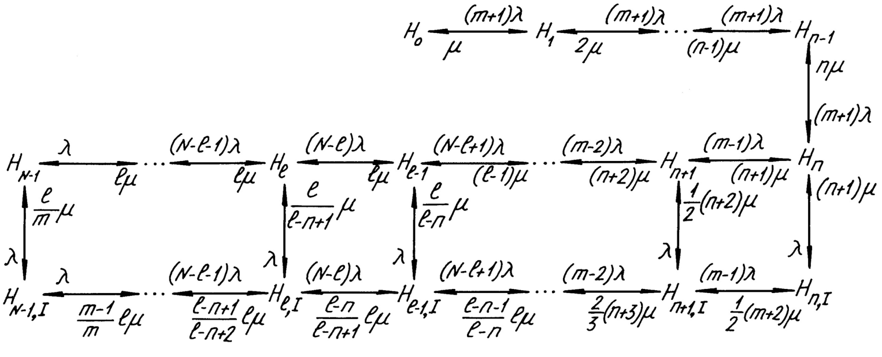


Fig. 4. Markov chain state transition diagram $(n \le l \le N)$

4 Numerical Example

Let us investigate the reliability of selected communication channel with common set of redundant radio stations and with dynamic management of backup modes in ATM communication network. It is possible to evaluate the reliability in the proposed model with two-stage mode of reserve elements in comparison to the active backup mode of redundant elements with the help of the factor of reliability improvement V:

$$V = \frac{1 - A_1}{1 - A_2}$$

where the value of A_2 is determined in accordance with the expression (1), equation for the A_1 availability of the system with active backup mode of redundant elements was determined in [11].

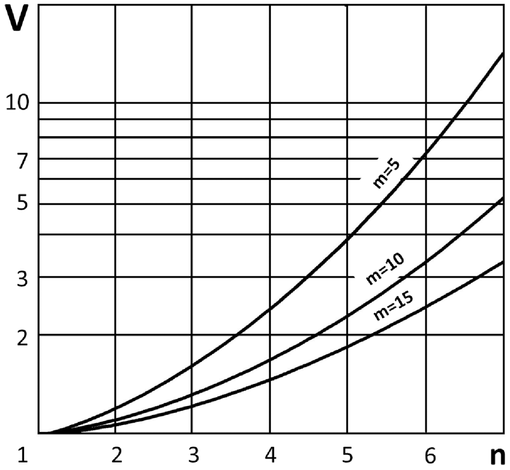


Fig. 5. The factor of reliability improvement

At the Fig. 5 the factor of reliability improvement V is shown as function of number m of communication channels with different number n of standby radio stations in common redundant set of RS for Mean Time Between Failures of each RS MTBF = 1/λ = 3000 h, l = 1 and μ = 2 h.

Analysis of the curves at Fig. 5 shows that factor of reliability improvement dramatically increases with increasing number of redundant elements at the second stage mode of reservation. The intensity of the given functional relation decreases with increasing number of channels.

5 Conclusions

Mathematical model of the channel reliability of the communication network of Air Traffic Management system in the real conditions of operation is developed. ATM systems represent essential infrastructure that is critical for flight safety and communication is a key element in the present ATM system. As additional method of improving the reliability of the ATM communication network a dynamic management of backup modes with two-stage mode of redundant elements is proposed. Mathematical model of the channel reliability is developed. Expressions for channel availability of the ATM communication network with common set of redundant radio stations and dynamic management of backup modes with two-stage mode of redundant elements are developed. Comparative analysis of redundancy effectiveness for developed and used structure of ATM communication network is performed.

It is shown that at the first stage two-stage mode of redundancy is optimal to have only one no-fault element. In this case channel reliability of communication network increases with increasing number of redundant elements at the second stage mode of reservation. This dependence is more active in the communication networks with a smaller number of channels.

Acknowledgment. This work was supported by Latvian state research programme project "The Next Generation of Information and Communication Technologies (Next IT)" (2014–2017).

References

1. Wiegmann, D., Shappell, S.: Human error perspectives in aviation. Int. J. Aviat. Psychol. **11** (4), 341–357 (2001)
2. Ireson, W., Coombs, C., Moss, R.: Handbook of reliability engineering and management 2/E, 2nd edn. McGraw-Hill Education, New York (1995). 816 p.
3. Modarres, M., Kaminskiy, M., Krivtsov, V.: Reliability Engineering and Risk Analysis: A Practical Guide. Quality and Reliability, 2nd edn. CRC Press, Boca Raton (2009). 470 p.
4. Barlow, R., Heidtmann, K.: On the reliability computation of a k-out-of-n system. Microelectron. Reliab. **33**(2), 267–269 (1993). Elsevier
5. Misra, K.: Handbook of Performability Engineering. Springer, London (2008). 1315 p.
6. McGrady, P.: The availability of a k-out-of-n:G network. IEEE Trans. Reliab. **R-34**(5), 451–452 (1985)

7. Liu, H.: Reliability of a load-sharing k-out-of-n:G system: non-iid components with arbitrary distributions. IEEE Trans. Reliab. **47**(3), 279–284 (1998)
8. Rushdi, A.: A switching-algebraic analysis of consecutive-k-out-of-n:F systems. Microelectron. Reliab. **27**(1), 171–174 (1987). Elsevier
9. Ayers, M.: Telecommunications System Reliability Engineering, Theory, and Practice. Wiley-IEEE Press, Hoboken (2012). 256 p.
10. Chatwattanasiri, N., Coit, D., Wattanapongsakorn, N., Sooktip, T.: Dynamic k-out-of-n system reliability for redundant local area networks. In: 2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1–4, 16–18 May 2012
11. Kozlov, B., Ushakov, I.: Reliability Handbook. International Series in Decision Processes. Holt, Rinehart & Winston of Canada Ltd, New York (1970)
12. Kabashkin, I.: Effectiveness of redundancy in communication network of air traffic management system. In: Dependability Engineering and Complex Systems. Advances in Intelligent Systems and Computing, Proceedings of the Eleventh International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, June 27–July 1 2016, Brunów, Poland, vol. 470, pp. 257–265. Springer, Switzerland (2016)
13. Kabashkin, I:. Analysing of the voice communication channels for ground segment of air traffic management system based on embedded cloud technology. In: Information and Software Technologies. Communications in Computer and Information Science, Proceedings of the 22nd International Conference, ICIST 2016, Druskininkai, Lithuania, 13–15 October 2016, vol. 639, pp. 639–649. Springer International Publishing, Switzerland (2016)
14. Raikin, I.: Elements of Reliability Theory for Technical Systems. Sov. Radio Publisher, Moscow (1978). 280 p. (in Russian)
15. Rubino, G., Sericola, B.: Markov Chains and Dependability Theory. Cambridge University Press, Cambridge (2014). 284 p.

Availability Models and Maintenance Strategies for Smart Building Automation Systems Considering Attacks on Component Vulnerabilities

Vyacheslav Kharchenko[1,2], Yuriy Ponochovnyi[3](✉),
Al-Sudani Mustafa Qahtan Abdulmunem[1], and Anton Andrashov[2]

[1] National Aerospace University KhAI, Kharkiv, Ukraine
V.Kharchenko@csn.khai.edu, mostafahkahtanl@gmail.com
[2] Research and Production Company Radiy, Kropyvnytskyi, Ukraine
a.andrashov@radiy.com
[3] Poltava National Technical University, Poltava, Ukraine
pnchl@rambler.ru

Abstract. The paper deals with developing and researching Markov's models and assessing of the availability of Instrumentation and control systems which are a part of smart building automation system (BAS). It was determined that the causes of failures and unavailability of the BAS component architecture can be hardware (physical), software (design) faults, and successful attacks on vulnerabilities (interaction faults), first of all. BAS failures are related to reliability issue, attacks on vulnerabilities is related to security issue. These two reason groups are considered as elements of two disjoint sets. The paper presents the detailed analysis of the BAS architecture consisting of control (FPGA-based), communication (ZigBee) and data levels considering their faults and vulnerabilities. Besides, maintenance procedures (without, common and separate maintenance for reliability and security) are described.

Keywords: Software faults and vulnerabilities · Availability · Maintenance strategy · Markov models · Smart buildings · Building automation system

1 Introduction

The development of virtualization technology and the creation of cloud computing environments are responsible for the appearance of new variants of the architecture of IT systems, which must be considered when assessing and ensuring the quality of modern computer systems and services, which include a system of "smart home". This dynamic character of the processes of information interaction significantly complicates the possibility of rapid assessment of the reliability and availability of software and infrastructure resources available to remote access [1].

According to the international and national standards in [2, 3], we can assess the level of risk for a building automation system and give the requirements that must be met to achieve the desired goal of safety and availability.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_18

The primary goals of the work in [4] are security issues for system design and the integration of security subsystems, which significantly tightens security requirements to the protocol of a network control system, and weaknesses in the system design according to hardware and software components.

In [5] it deals with development and research of Markov models of smart building automation systems (BAS), it has been taken into account that BAS failures can be caused by intra (reliability) and external (security) reasons including software faults and attacks on vulnerabilities. The sets of faults and vulnerabilities are considered as separated and disjoint ones, Markov models of BAS architecture with occurred software faults and attacked vulnerabilities considering three maintenance strategies are systemized and researched. These strategies are based on recovery without maintenance, maintenance with common and separate activities on reliability (faults) and vulnerabilities (security). Recommendations concerning choice of strategies and parameters of maintenance are suggested.

We develop number of strategies using a Markov model. These strategies deal with the system availability; it describes the possibility to recover the system from the down state (the state when there is a need to use these strategies) to the up state (the level of availability according to the customer requirements). The architecture of these strategies depends on the kind of maintenance (common or separate). The result of these strategies give different ways for recovering the system taking into account the customer requirements as the maximum value of availability during the minimum time.

In Sect. 2 we presented the analysis of system design and presented the classification of BAS. In Subsect. 2.1 the architecture of BAS and the main components in system design are analyzed. The faults and vulnerabilities of the main BAS components (FPGA, ZigBee, and database) are described in Subsect. 2.2. Subsection 2.3 analyzes the model and tree to draw the structure of the steps of the analysis using the Markov model and give the wide picture of system analysis. Subsection 2.4 presents the review of the development of Markov model strategy and how it can be applied to this work, and describe the parameter of the strategy. In Sect. 3 the analysis of system using Markov model for separated maintenance strategy is performed via marked graphs for the model of separated maintenance [5]. Section 4 is dedicated to research of Markov models depending to use the different strategies.

2 Approach and Modeling Technique

Analysis of the system is performed to determine its dependability taking into account reliability and security issues; in this work we have developed a number of strategies, which are used to analyze complex and big systems. According to [6, 7] BAS design have been divided into three levels, system availability will depend on these levels. The analysis the components for each level in [8] is performed. In Fig. 1, the classification of BAS describes the parameter of system design, which is divided between the reliability issues and security issues.

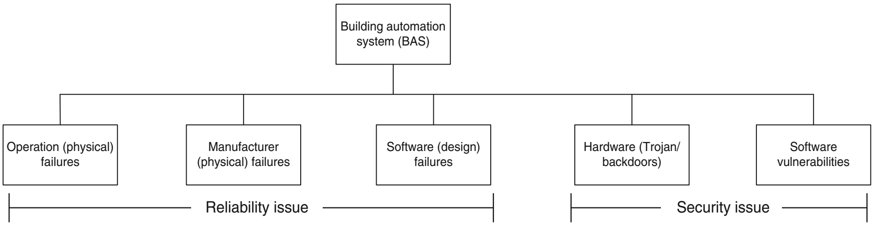

Building automation system (BAS)
Operation (physical) failures
Manufacturer (physical) failures
Software (design) failures
Hardware (Trojan/ backdoors)
Software vulnerabilities
Reliability issue
Security issue

Fig. 1 Analysis the availability of building automation system

2.1 Building Automation System Architecture and Components

BAS components are different depending on the area of system application, but in general they can be divided as:

1. Upper level (Management Level): dispatching and administration as well as work with databases and statistical functions. At this level cooperation between personnel (operators, dispatchers etc.) and system is performed, which is implemented by means of computer devices and SCADA systems. In our case study, we analyze the database of this level taking into account reliability and security.
2. Middle level (Communication Level): it is responsible for connection between levels and sending/receiving the information. According to our analysis we choose ZigBee as one of these level components.
3. Low level (Automation Level): level of terminals with input/output functions. This level includes sensors, actuating mechanisms, cabling between devices and low middle levels. One of the important components used for this level is FPGA [1].

These levels are divided depending on our vision of analysis for the system as shown in Fig. 2. There are different designs of BAS but we choose this design as the easiest in use and analysis.

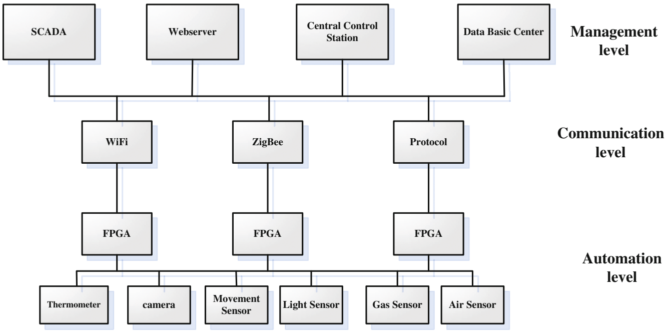


Fig. 2 Architecture of building automation system

2.2 Component Faults and Vulnerabilities

Field programmable Gate Arrays (FPGAs) are silicon devices, which are ready to be used. They can be electrically programmed and then can be used as a kind of system or digital circuit. One of the features of FPGAs is easiness of configuration and cost-effectiveness. It is also possible to make any updates and upgrade it. To do this it is necessary just to download a new application bit stream. FPGAs have numerous advantages but nevertheless, design flexibility remains their main advantage, when we consider cyber-security of FPGA we must take into account all parts involved in the life cycle of the FPGA chips and FPGA-based BAS. These are an FPGA chip vendor, a developer of the BAS as well as a user of FPGA-based BAS. The analysis of cyber-security for FPGA technology includes the development process as well as the operation of the integrated BAS. It must be noted that cyber-security vulnerabilities can be introduced by:

- the FPGA-chip vendor, during designing, manufacturing, packing and testing of FPGA chips;
- the BAS developer, i.e. when FPGA electronic design is developed, implemented or rested; the operator of the BAS, i.e. it is possible to make changes in the operating BAS during operation or maintenance.

Database attacks have increased because of the increased availability of access to data stored in those databases, database in BAS design contents the information, which is important for the system and data from different levels for management and storage, when the access to the stored information will be available for several users, it will increase the possibility of data theft, that is why it is necessary to control this kind of access because in the BAS system the attacker aims to access the important information, which he can use for attacks or monitoring the system.

Let us consider what wirelesses networks consist of they have four basic components. These are: the transmission of data via radio frequencies; access points providing a connection to the organizational network and/or the client devices (laptops, PDAs, etc.); and Users, the given components may have vulnerabilities and be attacked and this will result in the compromise of confidentiality, integrity, and availability.

2.3 General Models

The pervious analysis in [8] was related using ATA technique taking into account the vulnerabilities of system components. To analyze BAS and system components and receive the state probability we need to use the Markov model. In the Markov model [9] we have possibility to add more components and eliminate them without any effect on the analysis process. During the first step in analysis process we need to give a big picture for system with all possible sates, which the system can be in throughout its cycle life.

Use Markov model to draw the path of transmission of system without taking into account the past of state, also we have possibility to add more sate path to system analysis without need to change the another state, the analysis will be divided between

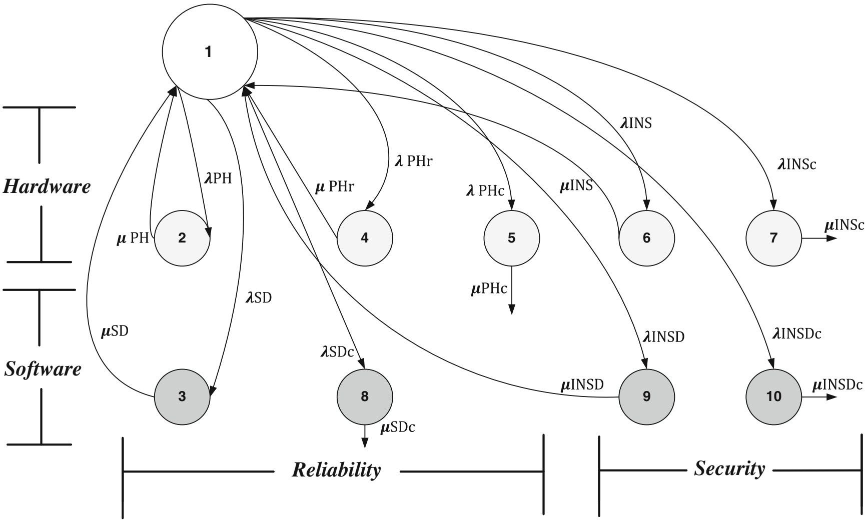


Fig. 3 Markov graph for BAS availability

use the (separated and common maintenance). In Fig. 3, the first model, which analyzes all the possible states of the system and shows the transmission between state and recovery. In Table 1 it describes transmission and recovery between the states. When the Markov model is developed, we have eliminated number of states as shown in Fig. 4a (first simplification).

Table 1 Description of parameters Markov graph

Parameter	Failure/recovery rates
λPH	Physical operation failure (hardware)
μPH	Physical operation failure (hardware/repair)
λPHr	Physical failure operation (soft error)
μPHr	Physical operation failure (soft hardware error/restart)
λPHc	Physical manufacture failure (hardware)
μPHc	Manufacture failure (hardware/changing design)
λINS	Intrusion failure (soft hardware vulnerability)
μINS	Intrusion failure (soft hardware vulnerability/restart)
λINSc	Intrusion failure (severe hardware vulnerability)
μINSc	Intrusion failure (severe hardware vulnerability/changing design)
λSD	Failure caused by design fault (software)
μSD	Soft error caused by design fault (software/restart)
λSDc	Failure caused by design fault (software)
μSDc	Failure caused by design fault (software/changing code)
λINSD	Intrusion failure (soft software vulnerability)
μINSD	Intrusion failure (soft software vulnerability/restart)
λINSDc	Intrusion failure (severe software vulnerability)
μINSDc	Intrusion failure (severe software vulnerability/changing code)

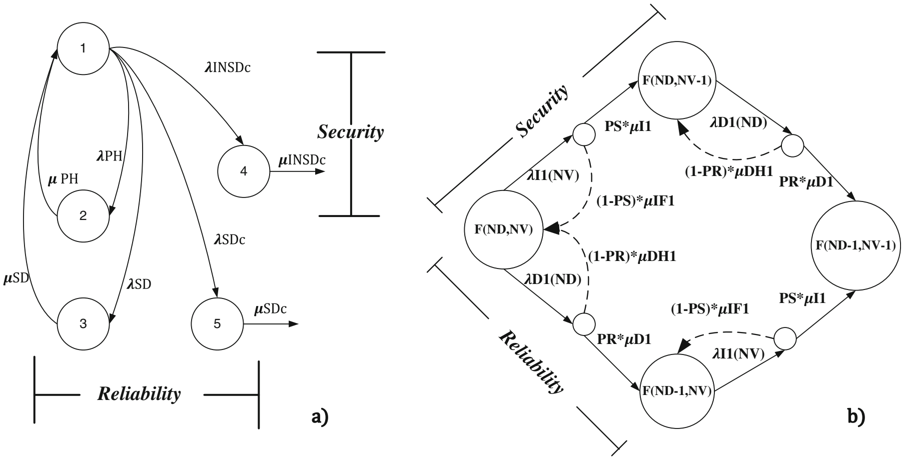


Fig. 4 Simplified Markov model

Figure 4b shows the second simplified Markov graph. Second simplification is the assumption of the successive manifestation of faults and vulnerabilities of component BAS. Also, it is assumed that a fault or vulnerability will be eliminated from the probability PR (PS).

2.4 Model Specification

We develop a few Markov model as shown in Table 2. The BAS analysis is divided into security issues and reliability issues. The sates of transmission for Markov model is divided according to these two issues. First, the security part is presented as Nv (number of vulnerability); second is the reliability Nd (number of faults). The goal of these model is to eliminate Nv, Nd in the minimum time of the system life cycle, and recover to the maximum value of availability (A_{MBAS}const) during period of time (T_{MBAS}const). In some cases, the elimination process inside the system will not be able to eliminate the vulnerability or design fault; in this case we add the maintenance strategies, which give the support for system to increase the elimination process. In our case we use two types of maintenances strategies:

(1) the common maintenance, which deals with design fault and vulnerability in same time, and it means that the process of elimination will be sequential between design fault and vulnerability;
(2) the separated maintenance, which deals with vulnerability and design fault separately one by one. In next section, we will be describing the characteristics of maintenance strategies for two models: one with common maintenance and another with separated maintenance.

Table 2 Basic models BAS

Model name	The number of faults	The number of vulnerabilities	Number of maintenance	Type of Maintenance
MBAS1	0……. Nd	0…..Nv	0	–
MBAS2	0……. Nd	0…..Nv	∞	Common
MBAS3	0……. Nd	0…..Nv	∞	Separate
MBAS4	0……. Nd	0…..Nv	0…..Np	Common
MBAS5	0……. Nd	0…..Nv	0…..Ndp, 0…. Ndv	Separate

3 Markov Model for Separate Maintenance

The model is also extended toward the base and includes additional separate state maintenance procedures, the number of maintenance states doubled since considered maintenance procedures, the purpose of which is to identify software faults only, and on the other hand, only vulnerabilities. Marked graph is shown in Fig. 5.

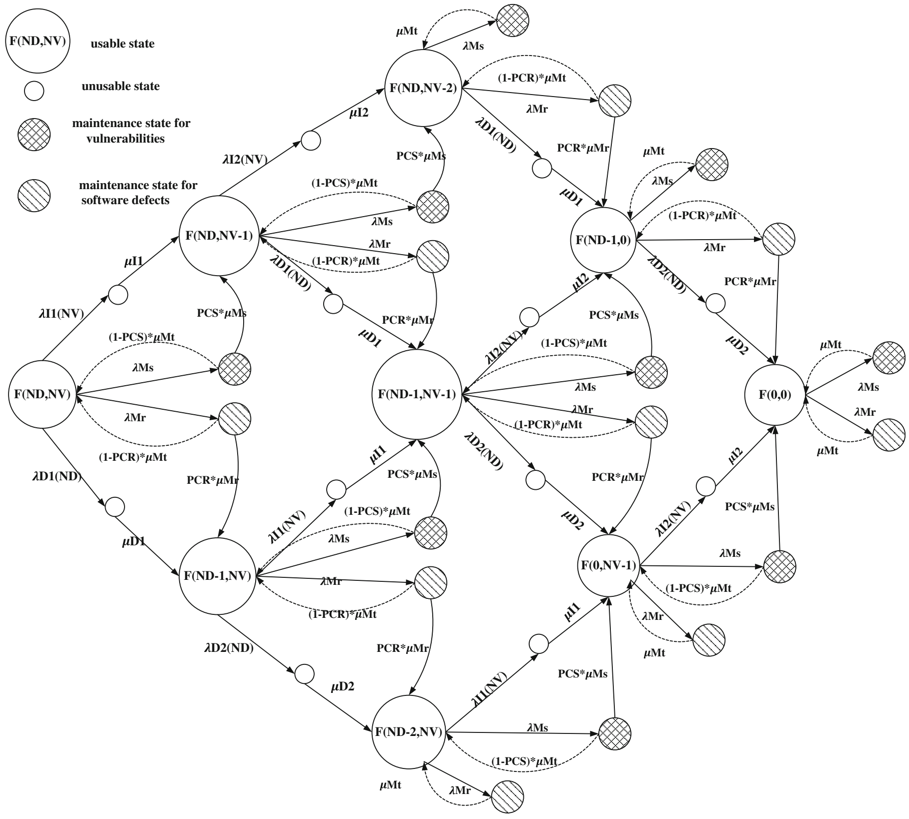


Fig. 5 Marked graph MBAS3 model for separate reliability/security maintenance

Conditions modeling separate maintenance procedures are shown by the dashed circles with different filling. Transitions in the maintenance of the state carried out a usable state: the maintenance state for vulnerabilities - with the intensity of maintenance λMs, in the maintenance state for software faults - with intensity λMr. Since we are considering separate maintenance, formed two complete groups of events: the identification of vulnerabilities in the maintenance with the probability of not detecting vulnerabilities and PCS with probability (1−PCS); identify software faults in the maintenance with the PCR probability and failure to identify faults with a probability of (1−PCR).

Each maintenance state is performed by two transitions vulnerabilities: the first - with PCS*μMs intensity simulates the detection and elimination of vulnerabilities for maintenance, the second - with the intensity (1−PCS)*μMt simulates holding maintenance without identifying vulnerabilities. In the case of the removal of all vulnerabilities transition from the maintenance weighted intensity μMt.

Similarly, there is simulation of transitions from maintenance state on software faults. Transitions with PCR*μMr intensity model identification and removal of software faults for maintenance, transitions with intensity (1−PCR)*μMt model holding maintenance without identifying faults.

4 Simulation and Comparative Analysis

Values of parameters for simulation were chosen the following as shown in Table 3.

Table 3 Values of input parameters of simulation processing

Symbol	Illustration	Value	Unit
laR(1)	The intensity of the first fault manifestation BAS λD1	5e−4	1/hour
laR(2)	The intensity of the second fault manifestation BAS λD2	4.5e−4	1/hour
laS(1)	Intensity of the first vulnerability manifestation BAS λI1	3e−3	1/hour
laS(2)	The intensity of the second vulnerability BAS λI2	3.5e−3	1/hour
muR(1)	The intensity of the restoration with the removal of the first fault BAS μD1	0.5	1/hour
muR(2)	The recovery rate with the elimination of the second fault BAS μD1	0.4	1/hour
muS(1)	The recovery rate with the removal of the first vulnerability BAS μI1	0.45	1/hour
muS(2)	The recovery rate with the elimination of the second vulnerability BAS μI2	0.34	1/hour
muRH	The intensity of the restart without removing faults μDH1 = μDH2	5	1/hour
muSF	The intensity of the restart without removing vuln. μIF1 = μIF2	6	1/hour
PR	The probability of fault elimination of the BAS during recovery	0.9	
PS	The probability of eliminating the vulnerability of the BAS during recovery	0.9	
Nd	The number of faults in the system BAS	2	
Nv	The number of vulnerabilities in the system BAS	2	
laMj	The intensity of the common maintenance λMj	1e−3	1/hour
laMs	The intensity of the maintenance separate in vulnerabilities λMs	5e−3	1/hour
laMr	The intensity of the maintenance separate in faults λMr	1e−3	1/hour
muMt	The intensity of holding measures on common maintenance μMt	0.4	1/hour
muMs	The intensity of detecting and removing a vulnerability μMs	0.2	1/hour
muMr	The intensity of detecting and removing a fault μMr	0.3	1/hour
PCS	The probability of identifying vulnerabilities in the maintenance process	0.4409	
PCR	The probability of identifying a software fault in the maintenance process	0.388	

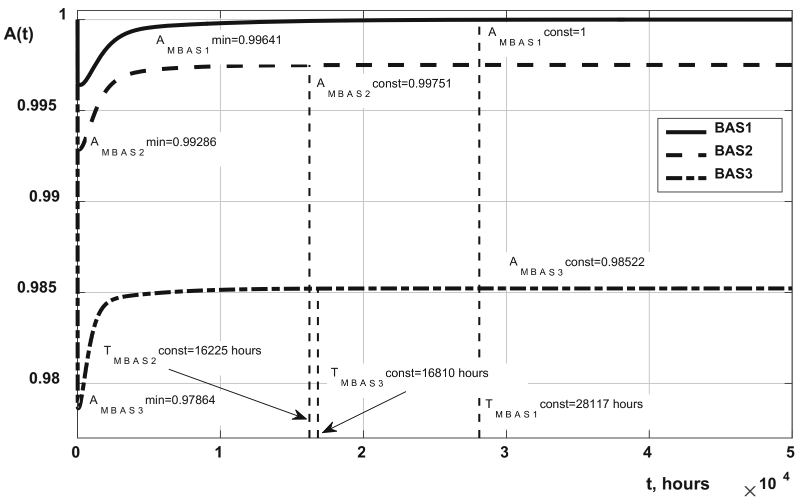


Fig. 6 Simulation results for BAS availability models

Results of simulation are illustrated by Fig. 6. The following conclusions can be formulated according with simulation of BAS with two different strategy of maintenance.

Graphs of availability function have the same view (Fig. 6): at the first stage availability is decreased to minimum and then is increased to stable value. Hence, three parameters should be taken into account:

- a value of availability function minimum A_{MBAS}min (for the model MBAS1 – 0.99641, the model MBAS2 – 0.99286, for the model MBAS3 – 0.97864);
- a value of availability function in stable mode A_{MBAS}const (for the model MBAS1–1, for the model MBAS2 – 0.9975, for the model MBAS3 – 0.9852);
- time of transition in stable mode T_{MBAS}const (for the model MBAS1–28117 h, for the model MBAS2 – 16225 h, for the model MBAS3 – 16810 h).

5 Conclusions

The paper analyzes Markov models which are used to research smart building BAS availability and security considering maintenance strategies. The main strategies are based on common and separated maintenance and the possibility to recovery system by elimination the vulnerabilities and faults. Figure 6 shows the simulation of two strategies and the difference between the time to recovery and the maximum value of availability, the choosing of these strategies is done according to customer requirements.

Future steps include:

- development of integrated strategies for BAS maintenance oriented at Cloud Computing taking into account reliability and security policies;
- investigation of the impact on the availability and the safety of other types of BAS vulnerabilities.

References

1. Ian, K., Tessier, R., Rose, J.: FPGA architecture: survey and challenges. Found. Trends Electron. Des. Autom. **2**(2), 135–253 (2008)
2. ISO/IEC 15408-2:2010 Information technology - Security techniques - Evaluation criteria for IT security - Part 2: Security functional components. European Committee for Electrotechnical Standardization, Brussels (2008)
3. STO NOSTROY 2.35.73:2012 Engineering networks of high-rise buildings. Integrated safety maintenance systems of high-rise buildings and structures. National Association of Builders, Moscow (2014) (in Russian)
4. Granzer, W., Kastner, W., Neugschwandtner, G., Praus, F.: Security in networked building automation systems. In: IEEE International Workshop on Factory Communication Systems, Torino, Italy, pp. 283–292 (2006)
5. Kharchenko, V., Ponochovnyi, Y., Mustafa Qahtan Abdulmunem, A.-S.: Markov models availability of information and control system of smart buildings with separate and common maintenance in terms of reliability and security. Syst. Control Navig. Commun. **4**(36), 88–94 (2015)
6. Boyanov, L., Zlatogor, M.: Cyber security challenges in smart homes. In: Cyber Secur. Resiliency Policy Framework, pp. 99–114 (2014)
7. Huffmire, T., et al.: Managing security in FPGA-based embedded systems. IEEE Des. Test Comput. **25**(6), 590–598 (2008)
8. Mustafa Qahtan Abdulmunem, A.-S., Al-Khafaji Ahmed, W., Kharchenko, V.: ATA-based security assessment of smart building automation systems. Radioelectronic Comput. Syst. **3** (77), 30–40 (2016)
9. Trivedi, K.S. Kim, D.S., Roy, A., Medhi, D.: Dependability and Security Models. In: Proc. 7th International Workshop on the Design of Reliable Communication Networks (DRCN 2009), pp. 11–20. Washington, DC, USA (2009)

Concept of Multi-criteria Evaluation of the Airport Security Control Process

Artur Kierzkowski and Tomasz Kisiel(✉)

Faculty of Mechanical Engineering, Wroclaw University of Science
and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{artur.kierzkowski,tomasz.kisiel}@pwr.edu.pl

Abstract. The aim of the article was to develop a concept of a model that allows for assessing the security control process, taking into account the evaluation indices of security, capacity and the level of service. For this purpose, a hybrid model was proposed which used the computer simulation and fuzzy logic method. The performed analysis allows for identifying relationships between individual indicators and specification of the influence of one of the indicators on the other ones. The proposed model extends the existing approach to the security control process evaluation that was based on the independence of the aforementioned evaluation criteria. Only one selected factor was considered. The proposed model allows for selecting an appropriate structure of the technical system and the structure of the process including the criteria of security, capacity and level of service.

Keywords: Security control · Airport · Multiple-criteria evaluation

1 Introduction

The Commission Implementing Regulation No. (EU) 2015/1998 imposes on airport managers the obligation to designate boundaries between the landside and the airside. The access to the landside area of an airport is possible after checking authorized persons and after a security check. The aim of the article was to develop a model that allows for assessing the security control process, taking into account the evaluation indicators of security (efficiency), capacity and the level of service.

The aim of the security control process is the prevention of forbidden objects specified [1]. The legislator allows for performing security control by means of various control methods. For example, passengers undergo security control using at least one of the following methods:

- a manual search,
- walk-through metal detectors (WTMD),
- dogs for detection of explosives,
- devices for detecting trace amounts of explosives (ETD),
- security scanners,
- devices for detecting trace amounts of explosives (ETD) combined with a hand-held metal detector (HHMD).

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_19

Various methods of control also apply to objects brought to the operating area. The legislator accurately specifies the possibility of using control methods; however, the process implementation methods are not defined in such a precise way. As a result, the process must be performed in various variants. Thus, it will be important for the airport operator to perform the process in accordance with security requirements while obtaining a specified efficiency of the system to ensure timely passenger service. The existing research on this matter concerns an analysis of these notions; however, they are described as separate issues.

Some scientific studies focus on examining the structure of the process and the structure of the system of the capacity of security control [7–9, 12, 23]. In such studies, an analysis of the sensitivity of the system on changes in selected parameters is performed (the number of operators, the volume of individual areas, the number of devices used etc.). This allows for designing an appropriate structure of the system. A comprehensive set of information on systemic analysis was presented in report [4].

Another group of scientific studies is devoted to issues related to the level of service. Such studies mostly focus on the process management method (management of the dynamic quantity of resources). The measure of the assessment are indicators characterizing the level of service – mostly the queuing time, system evaluation by passengers, functional availability of the system etc. There are a quite a few studies devoted to this aspect [10, 11, 13, 17, 26].

A separate group of studies focus on security issues. In this case, analyses are performed that concern the reliability of systems for detecting forbidden objects [3, 19–21]. These studies show the level of security a given structure of the system or individual technical devices are characterized by.

At present, no approach is available that would allow for simultaneous multi-criteria assessment of the analysed structure of the security control system and process. The airport manager must adopt a management strategy for the system that will guarantee reliability as regards forbidden objects and also will allow for avoiding delays in flight operations, thus guaranteeing a high level of passenger service. A multi-criteria assessment was already presented in other areas [2, 5, 6, 14–16, 18, 22, 24, 25, 27–30], which confirms that it is commonly used.

2 Concept of a Model of Multiple-Criteria Evaluation of the Security Control System

The proposed concept of a multi-criteria evaluation assumes the development of a model on the basis of which it is possible to determine evaluation indicators in a parallel manner: reliability of detection of forbidden objects (efficiency), system capacity and the level of service. Next, on the basis of the values obtained, the entire system is evaluated, which allows for selecting the best solution for the pre-defined boundary conditions.

As input data, the structure of the technical system of security control must be determined as well as the method of performing the security control process (the structure of the process). One should also define figures for the structure of the system and process for determining output data. The concept of the multi-criteria evaluation of

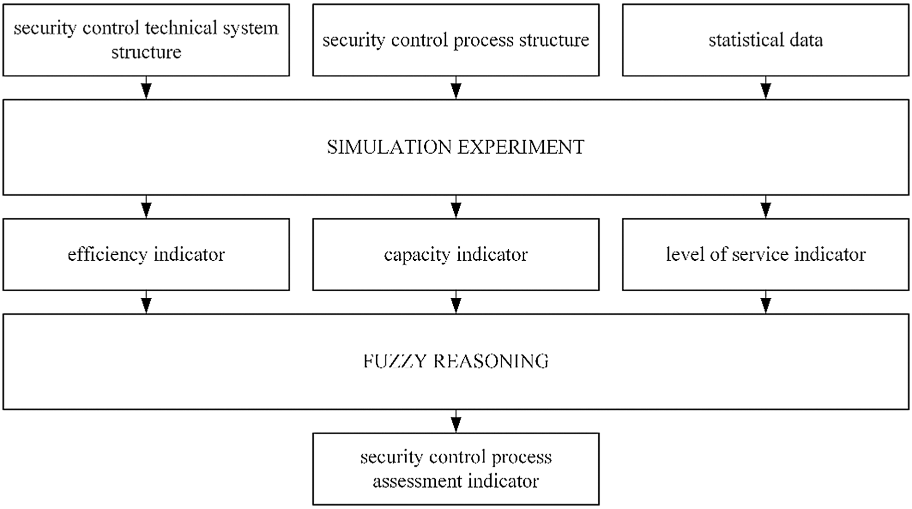

security control technical system structure
security control process structure
statistical data
SIMULATION EXPERIMENT
efficiency indicator
capacity indicator
level of service indicator
FUZZY REASONING
security control process assessment indicator

Fig. 1. Concept of a model of multiple-criteria evaluation of the security control process.

the security control system is presented in Fig. 1. It is based on a hybrid model using computer simulation methods and fuzzy logic.

For example, it was assumed that the security control process would be performed at a single security counter. In the system, there is a queue of passengers waiting for security control with an unlimited volume of $Q_1 = \infty$ passengers. The security checkpoint system consists of queues for WTMD control (metal detection) and x-ray control (detection using x-radiation). The volume of these areas is equal to $Q_2 = Q_3 = 8$ passengers. This means that 8 passengers submitting their objects for x-ray control can prepare for the WTMD control at the same time. It was assumed that all objects belonging to one passenger would be treated collectively as 1 piece of baggage. Detection of forbidden objects using a WTMD or x-ray device can be performed for 1 person/piece of baggage at the same time: $Q_4 = Q_5 = 1$. The system also includes a hand search and hand baggage control area. A passenger's hand search and hand baggage search is performed at the same time for 1 passenger/piece of baggage $Q_6 = Q_7 = 1$. At any given time, 5 pieces of baggage $Q_8 = 5$ and 2 passengers $Q_{10} = 2$ can be waiting for manual control of baggage. A passenger's objects that were not subjected to additional hand search are collected at the collection area, which can hold a total of 5 passengers $Q_9 = 5$. The diagram of a security counter is presented in Fig. 2.

The developed model performs the security process in a parallel manner for passengers in accordance with the algorithm we already presented in [10]. For the performance of subsequent activities in the process, we assumed theoretical random variables presented in Table 1.

Additionally, the time of a passenger's control with a WTMD device was assumed as deterministic equal 2 [s]. An appropriate probability of events was also adopted. The probability of a passenger being directed to the hand search was adopted as 0.1. The probability of detecting a metal objects with a WTMD device is equal to 0.99. The

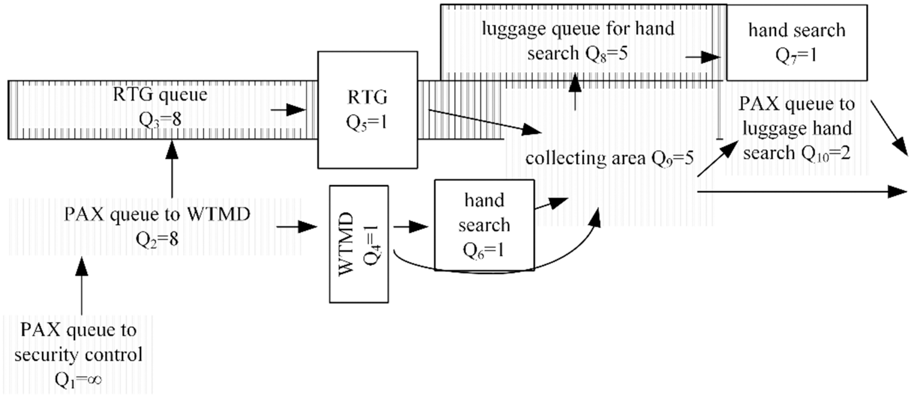

luggage queue for hand search Q_8=5
hand search Q_7=1
RTG queue Q_3=8
RTG Q_5=1
PAX queue to luggage hand search Q_{10}=2
collecting area Q_9=5
PAX queue to WTMD Q_2=8
WTMD Q_4=1
hand search Q_6=1
PAX queue to security control Q_1=∞

Fig. 2. Structure of the security control process

Table 1. Input data

Activity	Distribution function
Passengers inter-arrival time	Exponential (0.05)
Time of preparation for the control	Normal (60.20)
Time of x-ray control of the baggage	Normal (8.2)
Hand search time	Normal (20.5)
Time of manual control of baggage	Normal (30.10)
Baggage reclaim time	Normal (80.20)

Where: exponential(λ), normal(μ, σ), dimensions are in [s].

probability of detecting an object with an x-ray device is 0.80. The probability of baggage being directed to the hand search was adopted as 0.05. The probability of detecting forbidden objects by means of a hand search is 0.95. It was also adopted that in the level of service aspects, points are awarded on a 5 to 1 scale at an interval of 60 s. A passenger who waited in the queue for less than 60 s will give 5 points to the system. A passenger who waited in the queue from 60 to 120 s will give 4 points etc.

Performance of a computer simulation allows for determining the following indicators:

- the reliability indicator of forbidden object detection X_e – the share of correct detections in the stream of passengers holding a forbidden object;
- the capacity indicator of the security control system X_c – the average number of controlled passengers within an hour;
- the level of service indicator X_{LoS} – the average evaluation of the system depending on the passenger's queuing time.

Membership functions were assigned to input data from the simulation model to perform fuzzy reasoning. It was assumed that each variable could be represented by a low or high level. Membership function of the output linguistic variable is represented by a low,

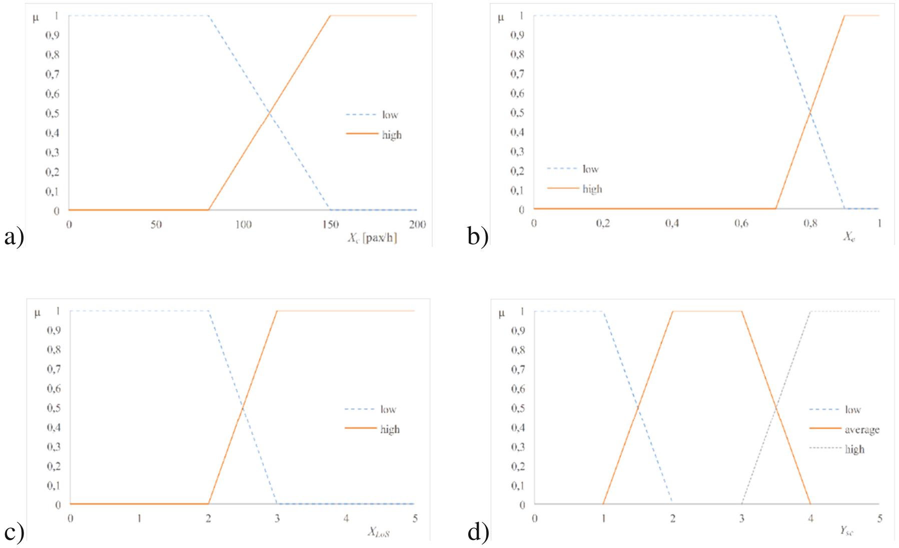


Fig. 3. Membership functions of the input and output linguistic variables (a) capacity of security control counter, (b) efficiency of prohibited items detection, (c) level of service, (d) evaluation of security control process.

average and high rate. Membership functions were developed on the basis of data presented in [4, 7] and the knowledge of experts from the Airport Wrocław (Fig. 3).

A set of rules was adopted to determine the index of the evaluation of the security control. In the presented example, the weight is the same for all rules. The set of rules has the following form:

- R1: If (X_c is low) and (X_{LoS} is low) and (X_e is low) then (Y_{sc} is low)
- R2: If (X_c is high) and (X_{LoS} is high) and (X_e is high) then (Y_{sc} is high)
- R3: If (X_c is high) and (X_{LoS} is low) then (Y_{sc} is average)
- R4: If (X_c is low) and (X_{LoS} is high) then (Y_{sc} is average)
- R5: If (X_c is low) and (X_e is high) then (Y_{sc} is average)
- R6: If (X_c is high) and (X_e is low) then (Y_{sc} is average)
- R7: If (X_{LoS} is low) and (X_e is high) then (Y_{sc} is average)
- R8: If (X_{LoS} is high) and (X_e is low) then (Y_{sc} is average)

For the adopted input data, FlexSim software was used in which a simulation model was used which allowed for the determination of input values for fuzzy reasoning. Next, an evaluation model was built on the basis of a fuzzy logic method using Matlab software.

The devices used for detecting forbidden objects often concern specific groups of objects. For example, the WTMD device is used for detecting metal objects only. Thus, the results obtained were presented in various groups depending on the type of forbidden object and the method of carrying it. The summary is presented in Table 2.

Table 2. Evaluation of the security control process

Type of forbidden object	XLoS	Xc	Xe	Ysc
Metal object carried by the passenger	4.95	140	0.99	3.78
Explosive object carried by the passenger	4.95	140	0.10	2.50
Metal object carried in the baggage	4.95	140	3.71	3.71
Explosive object carried in the baggage	4.95	140	3.71	3.71

For the adopted input data, the capacity of 140 passengers per hour was obtained. Such a system would be evaluated at 4.95 points by passengers considering the waiting time for a security check. The security technologies used, however, caused a difference in the value of the obtained efficiency indicator. The minimal value was obtained for detection of explosives carried by passengers. The maximum value of the *Xe* index was obtained for detection of any objects carried in baggage. Such a differentiation also contributed to the final value of the assessment of the *Ysc* process.

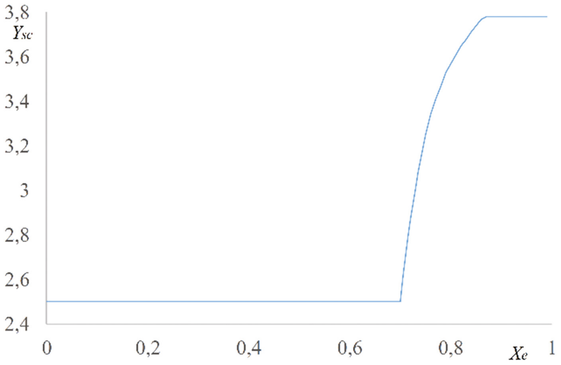


Fig. 4. Relationship between efficiency and the evaluation of the security control process.

Analysis of process sensitivity was conducted as regards the influence of the *Xc* indicator on the assessment of the *Ysc* process (Fig. 4.). It results from the conducted analysis that using various detection methods for objects and an attempt at changing procedures may influence the final evaluation of the system as regards $Ysc \in < 2.5, 3.78 >$. The maximum value of the index was obtained for detection of metal objects carried by passengers. The minimal value of process evaluation was obtained for the detection of explosives carried by passengers.

The analysis presented above was performed for theoretical input data. However, as shown, the presented concept of the model can be used for an actual system after performing actual research and implementing the results to the evaluation model.

3 Summary

The article presents a concept of the model of assessment of the security control process at the airport, which takes into account 3 indicators. The first of these is capacity. It is of strategic importance for the airport manager who strives to obtain the highest possible value of capacity. The efficiency indicator is an index opposite to capacity. Detection of forbidden objects is a key indicator for legislator and safety aspects. The use of additional control methods reduces capacity. Therefore, it is important to find an optimal solution guaranteeing punctual performance of the ground handling process while keeping an appropriate level of safety at the same time. The concept we present also allows for an evaluation of the system taking the aforementioned assumptions into account. The concept we present also takes into account a third index, which concerns the level of service. This, in turn, has a significant influence of the non-aeronautical revenue for the airport [10]. Thus, these are three key indicators that have not been considered simultaneously in scientific literature so far.

Further studies will be aimed at acquiring data from a real system and assessing various structures of the security control system and process to find the optimal solution.

References

1. EU, Commission Implementing Regulation (EU) 2015/1998 of 5 November 2015 laying down detailed measures for the implementation of the common basic standards on aviation security
2. Giel, R., Plewa, M.: The evaluation method of the process of municipal waste collection. In: CLC 2015: Carpathian Logistics Congress - Congress Proceedings. CLC 2015: Carpathian Logistics Congress - Conference Proceedings, pp. 281–287 (2016)
3. Hofer, F., Schwaninger, A.: Reliable and valid measures of threat detection performance in X-ray screening. In: 38th Annual 2004 International Carnahan Conference on Security Technology, pp. 303–308 (2004)
4. IATA, Security Access, Passenger Facilitation Campaign Results. International Air Transport Association (2012)
5. Jodejko-Pietruczuk, A., Nowakowski, T., Werbiska-Wojciechowska, S.: Issues of multi-stated logistic support system performing in a system of systems. In: Proceedings of CLC 2013: Carpathian Logistics Congress, Cracow, Poland, 09–11 December 2013, pp. 659–664 (2014)
6. Jodejko-Pietruczuk, A., Werbińska-Wojciechowska, S.: Block inspection policy for non-series technical objects. Safety, reliability and risk analysis: beyond the horizon. In: Proceedings of 22nd Annual Conference on European Safety and Reliability (ESREL) 2013, Amsterdam, pp. 889–898 (2014)
7. Kierzkowski, A., Kisiel, T.: Determination of the basic characteristics for security screening in winter air season, using simulation model of combined counter. Pr. Nauk. Politechniki Warszawskiej. Trans. **103**, 113–123 (2014)

8. Kierzkowski, A., Kisiel, T.: An impact of the operators and passengers' behavior on the airport's security screening reliability. In: Safety and Reliability: Methodology and Applications - Proceedings of the European Safety and Reliability Conference, ESREL 2014, pp. 2345–2354 (2015)
9. Kierzkowski, A., Kisiel, T.: Functional readiness of the security control system at an airport with single-report streams. In: Theory and Engineering of Complex Systems and Dependability. Advances in Intelligent Systems and Computing, vol. 365, pp. 211–221 (2015). doi:10.1007/978-3-319-19216-1_20
10. Kierzkowski, A., Kisiel, T.: Simulation model of security control system functioning: a case study of the Wroclaw Airport terminal. J. Air Trans. Manag. (2016, in press). doi:10.1016/j.jairtraman.2016.09.008
11. Kierzkowski, A.: Method for management of an airport security control system. Proc. Inst. Civ. Eng. Trans. **170**(1), 27–43 (2016). doi:10.1680/jtran.16.00036
12. Kirschenbaum, A.: The cost of airport security: the passenger dilemma. J. Air Trans. Manag. **30**, 39–45 (2013). doi:10.1016/j.jairtraman.2013.05.002
13. Manataki, I.E., Zografos, K.G.: Assessing airport terminal performance using a system dynamics model. J. Air Trans. Manag. **16**(2), 86–93 (2010). doi:10.1016/j.jairtraman.2009.10.007
14. Nowakowski, T., Tubis, A., Werbińska-Wojciechowska, S.: Maintenance decision making process - a case study of passenger transportation company. In: Theory and engineering of complex systems and dependability, pp. 305–318. Springer (2015). http://dx.doi.org/10.1007/978-3-319-19216-1_29
15. Restel F.J.: Defining states in reliability and safety modelling. In: Advances in Intelligent Systems and Computing, vol. 365, pp. 413–423 (2015). doi:10.1007/978-3-319-19216-1_39
16. Restel, F.J., Zajac, M.: Reliability model of the railway transportation system with respect to hazard states. In: 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 1031–1036 (2015)
17. Schultz, M., Fricke, H.: Managing passenger handling at airport terminals, In: Ninth USA/Europe Air Traffic Management Research and Development Seminar, Berlin, Germany (2011)
18. Siergiejczyk, M., Krzykowska, K., Rosiński, A.: Reliability assessment of integrated airport surface surveillance system, In: Proceedings of the Tenth International Conference Dependability and Complex Systems DepCoS-RELCOMEX, (red.) Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Advances in Intelligent Systems and Computing, vol. 365, pp. 435–443. Springer (2015). doi:10.1007/978-3-319-19216-1
19. Skorupski, J., Uchroński, P.: A fuzzy reasoning system for evaluating the efficiency of cabin baggage screening at airports. Transp. Res. Part C: Emerg. Technol. **54**, 157–175 (2015). doi:10.1016/j.trc.2015.03.017
20. Skorupski, J., Uchroński, P.: A fuzzy system to support the configuration of baggage screening devices at an airport. Expert Syst. Appl. **44**, 114–125 (2016)
21. Skorupski, J., Uchroński, P.: Managing the process of passenger security control at an airport using the fuzzy inference system. Expert Syst. Appl. **54**, 284–293 (2016)
22. Stańczyk, P., Stelmach, A.: Selected issues rules of weight control aircraft. KONES **23**(3), 496–503 (2016)
23. van Boekhold, J., Faghri, A., Li, M.: Evaluating security screening checkpoints for domestic flights using a general microscopic simulation model. J. Transport. Secur. **7**, 45–67 (2014). doi:10.1007/s12198-013-0129-8

24. Walkowiak, T., Mazurkiewicz, J.: Soft computing approach to discrete transport system management. In: Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence, vol. 6114, pp. 675–682 (2010). doi:10.1007/978-3-642-13232-2_83
25. Walkowiak, T., Mazurkiewicz, J.: Analysis of critical situations in discrete transport systems. In: Proceedings of DepCoS - RELCOMEX 2009, Brunów, Poland, 30 June–02 July 2009, pp. 364–371. IEEE (2009). doi:10.1109/DepCoS-RELCOMEX.2009.39
26. Wu, P.P.Y., Mengersen, K.: A review of models and model usage scenarios for an airport complex system. Transp. Res. Part A: Policy Pract. **47**, 124–140 (2013). doi:10.1016/j.tra.2012.10.015
27. Zajac, M., Swieboda, J.: Analysis of the process of unloading containers at the inland container terminal. In: Safety and Reliability: Methodology and Applications - Proceedings of the European Safety and Reliability Conference, ESREL 2014, pp. 1237–1241 (2015)
28. Zajac, M., Swieboda, J.: An unloading work model at an intermodal terminal. Adv. Intell. Syst. Comput. **365**, 573–582 (2015)
29. Zając, P.: Evaluation method of energy consumption in logistic warehouse systems. In: Environmental Issues in Logistics and Manufacturing. Springer (2015). http://dx.doi.org/10.1007/978-3-319-22044-4
30. Zając, P.: The idea of the model of evaluation of logistics warehouse systems with taking their energy consumption under consideration. Arch. Civ. Mech. Eng. **11**(2), 479–492 (2011). http://dx.doi.org/10.1016/S1644-9665(12)60157-5

Extending Continuous Integration with Post-mortem Debug Automation of Unhandled Exceptions Occurred in Kernel or User Mode Applications

Henryk Krawczyk and Dawid Zima(✉)

Department of Computer Systems Architecture,
Gdansk University of Technology, Gdańsk, Poland
henryk.krawczyk@eti.pg.gda.pl, dawid.zima@pg.gda.pl

Abstract. The paper proposes extension of the Continuous Integration practices with debug automation of unhandled exceptions. Goal of this improvement is to reduce the amount of redundant work when inspecting hundreds of failed tests from possibly the same reason, and to decrease time necessary to provide a fix to the codebase. The suitable CI infrastructure is proposed and an automatic method how to eliminate duplicated bugs is discussed. Also an example of such automation in the Windows operating system is presented.

Keywords: Continuous integration · Debug automation · Unhandled exceptions · Error reports · Statistic based debugging

1 Introduction

All software contains bugs. Some of them are detected during the stage of development (i.e. by running automated tests), but some of them are released with software to the users, and result in crashes. When the volume of crashes is very high, it becomes impossible to investigate all single occurrences of a bug, especially when the most of them are duplicates (single bug causing hundreds of clients or tests to crash). To facilitate this problem, it comes in handy to have a process that will help to remove duplicates, and group occurring problems into buckets representing bugs. The benefit of this approach is quite obvious: instead of investigating the cause of failure of each test, they are analyzed automatically and grouped into buckets, so the developer has to investigate a reduced number of buckets that has already been initially debugged by automates.

There can be distinguished 3 general use-cases for the debug automation systems: (1) handling error reports coming from the users after software release, (2) monitoring machine software and hardware configurations (i.e. corporate laptops/desktops in an organization, monitoring causes of server crashes etc.) and (3) extending Continuous Integration development process of complex systems.

The first use-case was well described by the Microsoft researchers from WER (Windows Error Reporting) [1] – also this system is described further in this paper. The

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_20

reason why special attention is paid to WER is because parts of it are used as a base in the example in this paper. More details are discussed in Sect. 4 of this publication.

The second use-case has been described and analyzed by the [2], analyzing causes of crashes of systems and applications of 200 research machines running Windows XP1 in the EECS department at UC Berkeley. It is worth to be mentioned, that data was gathered and analyzed using an internal version of WER system.

The main focus of this publication is gathered around the third use-case – the extension of Continuous Integration with a debug automation process – to answer the question what was the reason of a test failures. Most of the automatic tests that are part of the Continuous Integration process follow the “3-As” pattern (Arrange, Act, Assert). In “Arrange” phase all necessary inputs and preconditions are set. The “Act” phase invokes object or method under tests with data prepared in previous phase. The “Assert” phase checks, whether the output of the “Act” phase meets expected result. Also “Assert” is a phase, where test decides if it was failed or passed. But when something unexpected happens during “Act” stage, test also fails but not due to differences in expected output.

When test fails due to expected result mismatch it is often quite obvious why – so there should be no reason to automatically debug it. But when the test fails because of the unhandled exception resulting in application (or operating system) crash, the reason is often unknown and needs further investigation – so it is important to address such cases.

Other aspects related to the software development methods were discussed in previous publications [3]. Also, this publication emphasizes an innovative approach in Continuous Integration set of practices – to investigate not all failed tests, but automatically debugged reasons of those test failures caused by unhandled exceptions – other aspects of automations in software development process can be found in [4].

For the sake of this publication, the following definitions are used: error/exception – behavior of the computer program not intended by the programmer, unhandled exception – exception that was not handled by the program code causing it to crash, crash – unexpected program exit due to error/exception, bug/root cause – single cause of one or more errors in program code, bucket (noun) – error signature allowing one to group problems, bucket (verb) – process of classification error reports into buckets, call stack – ordered list of frames, in this case from the time when a crash occurred, order indicates relation between frames (caller), frame – consists of module and offset, often translated to function name using debug symbols, post-mortem debugging – process of finding root cause of the error after crash occurred using data collected during that crash, i.e. crash reports.

Section 2 of this publication contains description of the Continuous Integration set of practices including Continuous Delivery and Deployment, also with the extension of them with debug automation process. Section 3 discusses the idea of bucketing algorithms, Sect. 4 provides a complete example of the debug automation process in a Windows environment with description of the WER system. Section 5 concludes this publication, summarizing the automation process and emphasizing areas for development.

2 Continuous Integration and Automating the Debug Process of Unhandled Exceptions

As the basis of this paper are the practices of Continuous Integration, Delivery and Deployment, it is worth to shortly describe them and emphasize relations between them. More complex analysis of these (and other aspects of automations in software development process) can be found in another paper written by authors [4].

The Continuous Integration (CI) is a set of practices, known for a long time, but formally introduced as part of the eXtreme Programming methodology, and then well described by Martin Fowler [5]. He has distinguished the 11 most important practices of CI: (1) “Maintain a Single Source Repository”, (2) “Automate the Build”, (3) “Make Your Build Self-Testing”, (4) “Everyone Commits To the Mainline Every Day”, (5) “Every Commit Should Build the Mainline on an Integration Machine”, (6) “Fix Broken Builds Immediately”, (7) “Keep the Build Fast”, (8) “Test in a Clone of the Production Environment”, (9) “Make it Easy for Anyone to Get the Latest Executable”, (10) “Everyone can see what’s happening” and (11) “Automate Deployment”. It can be concluded in one sentence: the CI set of practices provides rapid feedback about committed change quality, and helps to avoid integration problems.

Continuous Delivery [6, 7] is the practice of developing software in a way, where it is always ready to be deployed to the production (software is deployable through its lifecycle and the development team prioritize keeping the software deployable over time spent working on a new feature). Continuous Delivery is built on the CI (adding stages responsible for deploying an application to production), so in order to do Continuous Delivery, you must be doing Continuous Integration. Continuous Deployment is a practice built on Continuous Delivery. Each change is automatically deployed to the production (which might result in multiple deployments per day). The main difference (and the only one) between Continuous Delivery and Continuous Deployment is that the deployment in Continuous Delivery depends on business decisions and is triggered manually, and in Continuous Deployment each “good” change (the one that has not broken the build and passed all of the tests) is immediately deployed to the production [6, 8].

When the developed application is very complex, consisting of many components with thousands of tests, sometimes information that the test failed may not be sufficient. Especially, when after the commit hundreds of tests start to fail at the same time. Inspecting all of them may be a time-consuming task. After all, it may be a single bug that caused multiple tests to fail.

As mentioned before, when a developer is receiving hundreds or thousands of error notices, it is almost impossible to investigate all of them, especially when they can have a single root cause (bug). Having a process that extends the default Continuous Integration process might be very crucial for complex systems, reducing time needed to fix the bugs, and also prioritize them.

The proposed method also reduces the chance of test noise (when unstable test or environment where application is being tested crashes but the cause is not in the application under test) influencing the results. The test noise can be explained using the example: tested application is a windows kernel mode driver - each application running in kernel mode shares a single virtual address space and because of this a crash in kernel mode results in an entire operating system crash. One of the test failure conditions might be detection of a system crash. But when the system crash occurred because of another faulty driver (not the tested one), the test will be marked as failed, but the reason of this failure was not related to the tested application – so noise was generated, and developers need to investigate this issue, wasting their time. Having automated the process of debugging, such cases can be quite easily filtered out, pointing to which test failures are noise-related.

Extension of CI process, with debug automation, has been presented on Fig. 1. Starting from the beginning, the developer commits his change to the source code repository. Then, CI server detects that new change was introduced to the repository and starts the build (compilation, static analysis etc.). When the compilation process is finished, automated tests are executed. Depending on the complexity of developed system, this step can be executed in hundreds of thousands of machines (physical or virtual). During this phase, due to errors introduced to the source code by developers, a tested application often crashes.

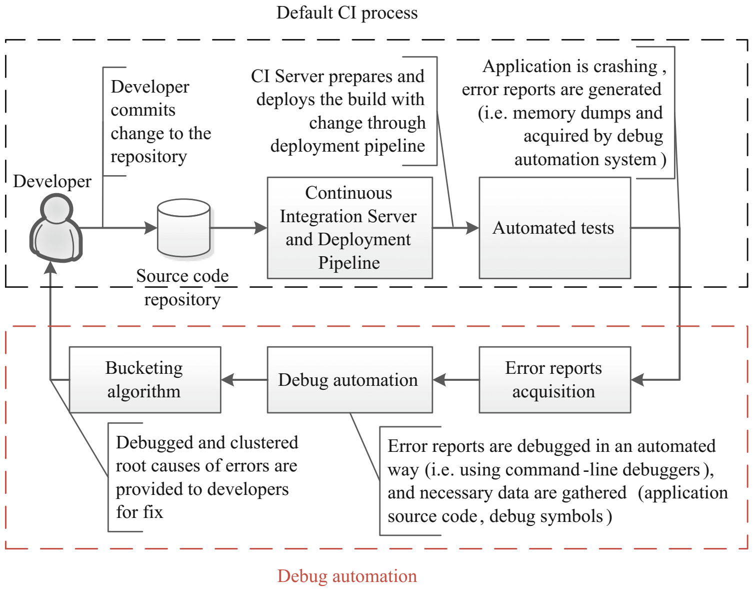

Default CI process
Developer
Developer commits change to the repository
Source code repository
CI Server prepares and deploys the build with change through deployment pipeline
Continuous Integration Server and Deployment Pipeline
Application is crashing , error reports are generated (i.e. memory dumps and acquired by debug automation system)
Automated tests
Error reports acquisition
Debug automation
Bucketing algorithm
Debugged and clustered root causes of errors are provided to developers for fix
Error reports are debugged in an automated way (i.e. using command -line debuggers), and necessary data are gathered (application source code , debug symbols)
Debug automation

Fig. 1. CI process with the proposed debug automation

In the proposed extension of default CI process, the Error reports acquisition stage defines how the error report is prepared when the crash occurred, and how it is sent to the next stage. The Debug automation is a stage, where from each acquired error report all necessary data is extracted (using debugger). The last stage, Bucketing algorithm, is to generate an appropriate bucket for the extracted debug data – the unique identifier of a bug.

After that, a report of failures is presented to the user. It contains not only information of which tests failed, but also the reason for those failures. It is very important, because hundreds of failed tests can have a single root cause, so the developer does not need to investigate each test, but only the reasons represented by the buckets. It significantly reduces the time to fix an issue in the source code.

The main advantage of the proposed debug automation process is a significant reduction of work and time needed to find and fix bug introduced to project source code during development having CI process set up. It's achieved by not analyzing all failed tests, but looking into distinct reasons of those test failures (which were debugged automatically) instead.

The implementation of debug automation extension in default CI process requires to solve some additional problems: method of error reports generation during the unhandled crash (currently all modern operating system can handle this case), method of error reports acquisition from test environment to debug servers or collecting additional debug artifacts from build stage needed by a debugger (i.e. debug symbols).

3 Bucketing Algorithms

Accuracy of the bucketing algorithm is crucial in debug automation systems analyzing high volumes of incoming data. The ideal bucketing algorithm should strictly maintain the property of orthogonality: one bug per bucket, and one bucket per bug [1]. "Second bucket problem" appears when multiple crashes caused by the same bug spread into more than one bucket, and "long tail problem" appears when multiple buckets containing a small number of crash reports, represents a single bug [1, 9].

In WER [1], all bucketing heuristics (client-side labeling and server-side classifying) have been divided into two main categories affecting the impact of a heuristic to a final bucket: expanding (to prevent assigning different bugs into one bucket, increases the number of buckets) and condensing (to assure that no two buckets represent the same bug, decreases the number of buckets). Also those two types of heuristics should not be conflicting with each other, instead they should be complimentary – expanding should not introduce new buckets for the same bug, and condensing should not put two bugs in one bucket [1].

In the server-side bucketing of WER, more than 500 heuristics have been implemented into windows debugger extension "!analyze" in 100 000 lines of code [1]. The most important ones are C1 to C5 – they allow the algorithm analyzing memory dump to identify thread context and stack frame which most likely caused the crash. Result of running "!analyze" command in windows debugger result in generating of a BUCKET_ID, i.e. the one presented on the Listing 1.

According to the [9], depending only on WER bucketing heuristics may result in a "long tail problem", thus there is a need of a more complex method. Another approach may be using the call stack of the thread when the crash happened - comparing error reports between each other based on a call stack similarity metric, and assigning them to one bucket when a threshold for similarity is reached.

Call stack is a stack (linear data structure of LIFO type – Last In, First Out) of actively executed program functions. On top of it there is a function that is currently executed by the operating system. The great advantage of the method comparing exceptions using call stack similarities is that it can be used to compare call stacks of a multi operating system application, allowing one to classify errors that happened on different operating systems (i.e. shared module used in Windows and Linux). This similarity may be computed using simple string-like similarity (i.e. Levenshtein distance) or a much more sophisticated method, like the one proposed by the Microsoft Research Team, Position Dependent Model (PDM) that's part of a more complex method called ReBucket [9]. However, those methods are complicated, including call stack pre-processing (i.e. for removing recurrences and "safe functions", unification of C++ templates etc.). However, sophisticated methods of generating failure buckets are very complicated problems and exceeds the scope of this publication.

4 Example

In the Windows operating system, the processor can run in two different modes: user and kernel [10]. All normal applications, and some of the drivers, run in user-mode, which means that when they are started, the Windows operating system creates the process for them with private virtual address space and the handles table. Because application virtual address space is private, one application cannot affect the data of another application. When the crash occurs, it is limited only to that application, other applications and operating system are not affected. Code running in kernel-mode shares a single virtual address space – this means that kernel-mode drivers are not isolated from each other and can damage each other and also the operating system, so if a kernel-mode driver crashes, the entire operating system crashes (displaying the "Blue Screen of Death" - BSOD).

Windows Error Reporting (WER) [1] is a distribute post-mortem debugging system developed by Microsoft and partially is a part of the Windows operating system. It originated from two initiatives from Microsoft developers: the Windows team developed a tool to automatically diagnose the most likely cause of the crash from a core dump, and an Office team tool to automatically collect mini dumps (stack trace with subset of heap memory) when the unhandled exception occurred. A combination of those two tools, automatic diagnosis with automatic error data collection, resulted in the creation of the WER service. The first program shipped with WER client side code was MSN Explorer.

The goal of the WER system, according to its authors, is to diagnose and correct every software error on every Windows systems. To achieve this goal, considering the enormous number of systems running Windows, WER must be scalable and able to deal with a big inflow of error reports every hour. So, to reduce the cost of error reporting when the volume is high, WER uses progressive data collection. Most of the buckets contain only a bucket identifier. When additional data is needed (i.e. this is the first signature of the problem), WER can collect additional information like mini dump, full memory dump or memory dumps of related processes. If the solution is already known for WER for the provided problem signature, the user is automatically provided the solution via a URL sent to him.

To achieve progressive data collection, WER uses a set of heuristics first on the client machine (called labeling) and then on the WER servers (called Classifying). Labeling, the client-side bucketing, is an important step, because in most cases only a bucket label will be sent to the WER servers.

To sum up the general WER system design: when the error condition is detected on the client machine, a special error with label bucket is generated and sent to the WER service. It stores the information about that error occurrence and, depending on the information it already has for the provided bucket label, it may request more details from the client machine (i.e. mini dump) or redirect a user or administrator of the client machine to an already known solution. When a crash occurrence for a bucket exceeds a threshold, WER automatically generates a bug report for developers in the bug tracker to investigate the issue [9]. According to the data collected and analyzed by Microsoft researchers [1], in comparison with errors reported by humans, bugs found by WER are 4.5 to 5.1 times more likely to be fixed.

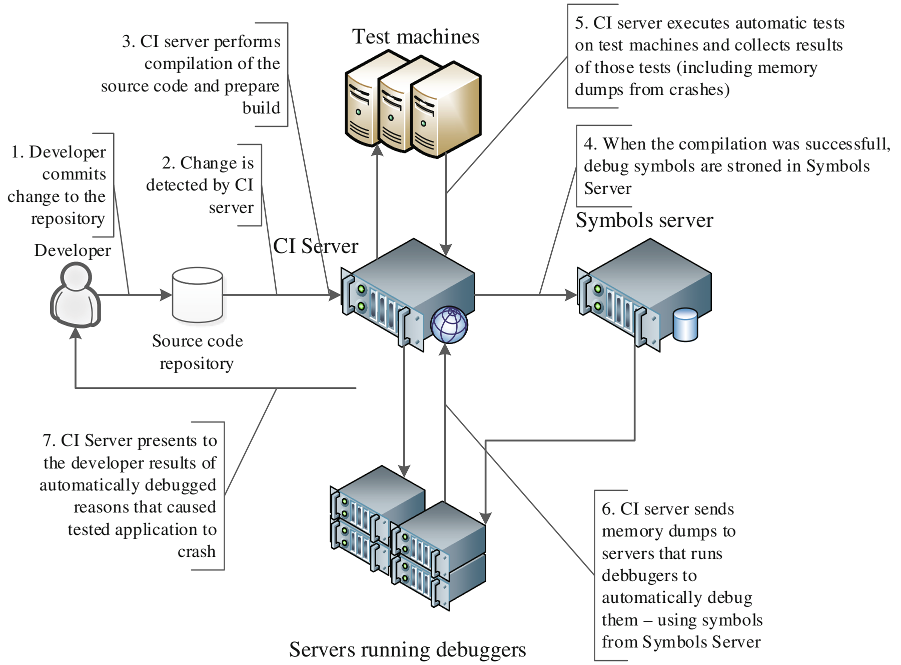
Despite WER there are some already existing systems for generating error reports, acquiring and debugging them, like Apple's CrashReporter for Mac OS X [11] or Mozilla's Crash Reports [12]. Also, all modern operating systems are capable of generating appropriate reports when the unhandled exceptions occur.

The Symbols files (.pdb) [13] in the Windows operating system are files generated by the compiler during the project build stage. They contain the following information: public symbols (typically all functions, static and global variables), a list of object files that are responsible for sections of code in the executable, frame pointer optimization information, name and type information for local variables and data structures, source file and line number information. They are used by the debugger to show developers more user-friendly information i.e. function names instead of offset value. They can be stored in Symbols Server by using the Microsoft Symstore utility.

A simplified example of a crashing Windows user-mode application will be considered. Figure 2 presents the entire infrastructure with all stages: (1) first, developer commits his change to the repository (in this example: missing NULL pointer check in function "GoLeft"), (2) committed change is automatically detected by the CI server (3) which performs compilation and building process. (4) After successful compilation debug symbols are added to the Symbols server (using Symstore utility). During the next step (5) CI server deploys the tested application to the test machines imitating production environment (but having set the registry key to create memory dumps for unhandled exception crashes) and perform tests. Some tests will fail due to missing NULL pointer check. Then, (6) CI server sends memory crash dumps to servers running debuggers (cdb.exe). The most important parts of debugger output have been presented on Listing 1. After that, CI server presents to the developer information about the results - in this example, an unhandled exception for the reason (missing NULL pointer check in "GoLeft" function) represented by the bucket from Listing 1. generated by WER heuristics in "!analyze" command of windows debugger. So, after that, the developer immediately can see what the root cause was of those failures, and quickly commit the fix. Handling crashes from kernel mode applications (drivers) in the Windows operating system is very similar – instead of using cdb.exe debugger, the kd.exe debugger should be used.

```
 [...]
BUCKET_ID:  NULL_POINTER_READ_CppConsoleApplication!BinaryTreeSearch::GoLeft+52
[...]
```

Listing 1. The most important part of debugger (cdb.exe) output after crash dump analysis



1. Developer commits change to the repository

2. Change is detected by CI server

3. CI server performs compilation of the source code and prepare build

4. When the compilation was successfull, debug symbols are stroned in Symbols Server

5. CI server executes automatic tests on test machines and collects results of those tests (including memory dumps from crashes)

6. CI server sends memory dumps to servers that runs debbugers to automatically debug them – using symbols from Symbols Server

7. CI Server presents to the developer results of automatically debugged reasons that caused tested application to crash

Fig. 2. Example of CI infrastructure with the debug automation extension

5 Conclusion

When the developed application is very complex (contains multiple modules, developed by many developers), benefits of extending the typical CI process by automating debugging are obvious – reduction of redundant work spend by programmers to narrow down the root causes of failing tests due to an unknown reason (unhandled exception causing crash), and shortening the time needed to fix errors in the application. The more the system is complex (contains more automated tests), the more advantages come from debug automation.

This paper discussed the Continuous Integration set of practices, and the problem of test failure due to unhandled exceptions, and proposed the solution with an example of implementation – the debug automation in Windows Operating system. The single bug introduced in a line of the code (missing NULL pointer check) caused different (and not relevant) tests to fail due to an unknown reason. Debug automation narrowed this down to a single root cause in the source code, saving time in debugging different cases with a common issue. Without this method, the developer would need to investigate manually (reproduce error in his development machine) those cases.

However, crucial in terms of debug automation is the bucketing algorithm. Its accuracy determines if developers will receive duplicated bug reports, or bug reports with multiple different problems aggregated wrongly into a single bucket. Microsoft researchers have done much work in this field [1, 9], but still there are many aspects that need additional researches.

For the sake of this paper, the scope of provided example was reduced to Windows operating system – this enables usage of heuristics implemented by Microsoft engineers in WER parts of the debuggers to generate a bucket (using "!analyze" command in KD or CDB debuggers). However, in some cases this approach is not sufficient [9] and there is a need to use more sophisticated methods, i.e. using call-stack similarities. Also, using call stack similarities methodology to compare error reports allows comparison of errors from different operating systems (for example when there is a shared module used by both, Windows and Linux applications). But dealing with call-stacks introduces other problems like how to calculate similarities, or how to treat recursions, "safe functions". Many of the aspects mentioned before will be the subject for further researches.

References

1. Glerum, K., Kinshumann, K., Greenberg, S., Aul, G., Orgovan, V., Nichols, G., Grant, D., Loihle, G., Hunt, G.: Debugging in the (very) large: ten years of implementation and experience. In: Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles (2009)
2. Ganapathi, A., Patterson, D.: Crash data collection: a windows case study Archana. In: International Conference on Dependable Systems and Networks (2005)
3. Zima, D.: Modern methods of software development. Task Q. **19**(4), 481–493 (2015)
4. Krawczyk, H., Zima, D.: Automation in software source code development. Int. J. Inf. Technol. Comput. Sci. (IJITCS) **8**(12), 1–9 (2016)
5. Fowler, M.: Continuous Integration. http://www.martinfowler.com/articles/continuousIntegration.html. Accessed 4 June 2016
6. Fowler, M.: ContinuousDelivery, 30 May 2013. http://martinfowler.com/bliki/ContinuousDelivery.html. Accessed 28 Nov 2015
7. Humble, J., Farley, D.: Anatomy of the deployment pipeline. In: Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, pp. 105–141. Addison-Wesley Professional (2010)
8. Humble, J.: Continuous Delivery vs. Continuous Deployment, 13 August 2010. http://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment/. Accessed 22 Nov 2015
9. Dang, Y., Wu, R., Zhang, H., Zhang, D., Nobel, P.: ReBucket: a method for clustering duplicate crash reports based on call stack similarity. In: Proceedings of the 34th International Conference on Software Engineering (2012)

10. User mode and kernel mode, Microsoft. https://msdn.microsoft.com/en-us/library/windows/hardware/ff554836. Accessed 9 Oct 2016
11. Technical Note TN2123 CrashReporter. https://developer.apple.com/library/mac/technotes/tn2004/tn2123.html. Accessed 5 June 2016
12. Mozilla Crash Reports. https://crash-stats.mozilla.com/home/product/Firefox. Accessed 5 June 2016
13. Microsoft, Debugging with Symbols. https://msdn.microsoft.com/en-us/library/windows/desktop/ee416588(v=vs.85).aspx. Accessed 15 Jan 2017

The Methodology of Studying of Active Traffic Management Module Self-oscillation Regime

Dmitry S. Kulyabov[1,2(✉)], Anna V. Korolkova[1], Tatyana R. Velieva[1],
Ekaterina G. Eferina[1], and Leonid A. Sevastianov[1,3]

[1] Department of Applied Probability and Informatics,
Peoples' Friendship University of Russia (RUDN University),
6 Miklukho-Maklaya Street, Moscow 117198, Russian Federation
{kulyabov_ds,korolkova_av,sevastianov_la}@rudn.university,
trvelieva@gmail.com, eg.eferina@gmail.com
[2] Laboratory of Information Technologies, Joint Institute for Nuclear Research,
6 Joliot-Curie Street, Dubna, Moscow Region 141980, Russian Federation
[3] Bogoliubov Laboratory of Theoretical Physics,
Joint Institute for Nuclear Research, 6 Joliot-Curie Street,
Dubna, Moscow Region 141980, Russian Federation

Abstract. Self-oscillating modes in computer networks control systems quite negatively affect the characteristics of these networks. The problem of finding the areas of self-oscillations is actual and important as the study of parameters of self-oscillations. Due to the significant nonlinearity of control characteristics, the study of the oscillatory modes presents certain difficulties. This paper describes the technique of research of self-oscillating modes on the basis of the control theory. This material is rather methodical than exploratory one.

Keywords: Traffic active management · Control theory · Self-oscillating mode

1 Introduction

While modeling technical systems with control it is often required to study characteristics of these systems. Also it is necessary to study the influence of system parameters on characteristics. In systems with control there is a parasitic phenomenon as self-oscillating mode. We carried out studies to determine the region of the self-oscillations emergence. However, the parameters of these oscillations were not investigated. In this paper, we propose to use the harmonic linearization method for this task. This method is used in control theory, but this branch of mathematics rarely used in classical mathematical modeling. The authors offer a methodological article in order to introduce this method to non-specialists.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_21

2 The RED Congestion Adaptive Control Mechanism

To improve the performance of the channel it is necessary to optimize the queue management at the routers. One of possible approaches is the application of the Random Early Detection (RED) algorithm (see [1,5,9,11,14]).

The RED algorithm uses a weighted queue length as factor determining the probability of packet drop. As the average queue length grows, the probability of packets drop also increases (see (1)). The algorithm uses two threshold values of the average queue length to control drop function (Fig. 1):

$$p(\hat{Q}) = \begin{cases} 0, & 0 < \hat{Q} \leqslant Q_{\min}, \\ \frac{\hat{Q} - Q_{\min}}{Q_{\max} - Q_{\min}} p_{\max}, & Q_{\min} < \hat{Q} \leqslant Q_{\max}, \\ 1, & \hat{Q} > Q_{\max}. \end{cases} \tag{1}$$

Here $p(\hat{Q})$ — packet drop function (drop probability), $\hat{Q}$ — exponentially-weighted moving average of the queue size average, $Q_{\min}$ and $Q_{\max}$ — thresholds for the weighted average of the queue length, $p_{\max}$ — the maximum level of packet drop.

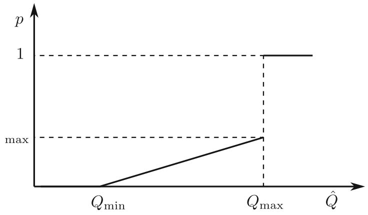


Fig. 1. RED packet drop function

The RED algorithm is quite effective due to simplicity of implementation in the network hardware, but it has a number of drawbacks. In particular, for some parameters values there is a steady oscillatory mode in the system, which negatively affects Quality of Service (QoS) indicators [10,15,19]. Unfortunately there are no clear selection criteria for RED parameters values, in which the system does not enter in self-oscillating mode.

To describe the RED algorithm we will use the following continuous model

$$\begin{cases} \dot{W}(t) = \frac{1}{T(Q,t)} - \frac{W(t)W(t-T(Q,t))}{2T(t-T(Q,t))} p(t - T(Q,t)); \\ \dot{Q}(t) = \frac{W(t)}{T(Q,t)} N(t) - C; \\ \dot{\hat{Q}}(t) = -w_q C \hat{Q}(t) + w_q C Q(t). \end{cases} \tag{2}$$

(see [4,6,7,12,13,16,17,20]) with some simplifying assumptions:

- the model is written in the moments;
- the model describes only the phase of congestion avoidance for TCP Reno protocol;
- in the model the drop is considered only after reception of 3 consistent ACK confirmations.

In (2) the following notation is used:

- W— the average TCP window size;
- Q— the average queue size;
- $\hat{Q}$— the exponentially weighted moving average (EWMA) of the queue size average;
- C— the queue service intensity;
- T— full round-trip time; $T = T_p + \frac{Q}{C}$, where T_p — round-trip time for free network (excluding delays in hardware); $\frac{Q}{C}$ — the time which batch spent in the queue;
- N — number of TCP sessions;
- p— packet drop function.

3 Harmonic Linearization Method

The method of harmonic linearization is an approximate method. It is used for study of start-oscillation conditions and determination of the parameters of self-oscillations, for the analysis and evaluation of their sustainability, as well as for the study of forced oscillations. Harmonically-linearized system depends on the amplitudes and frequencies of periodic processes. The harmonic linearization differs from the common method of linearization (leading to purely linear expressions) and allows to explore the basic properties of nonlinear systems.

We will use the block-linear approach in control theory [3]. According to this approach, the original nonlinear system is linearized and divided into blocks. These blocks are characterized by the transfer function linking the input and output values. The method of harmonic linearization is used for systems of a certain structure (see Fig. 2). The system consists of linear part H_l and the nonlinear part, which is set by function $f(x)$. It is generally considered a static nonlinear element.

For the harmonic linearization method free movement mode (input $g(t) = 0$) is assumed. The free harmonic oscillations are applied to the input of the nonlinear element:

$$x(t) = A\sin(\omega t). \tag{3}$$

On the output of the nonlinear element $f(x)$ we get a periodic signal. Let's expand it in a Fourier series:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_k \sin(k\omega t) + b_k \cos(k\omega t)), \tag{4}$$

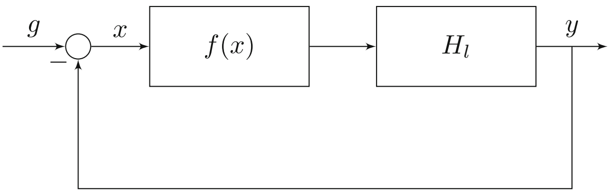


Fig. 2. Block structure of the system for the harmonic linearization method

where the coefficients of the Fourier series have the following form:

$$a_k = \frac{1}{\pi}\int_0^{2\pi} f(A\sin(\omega t))\sin(k\omega t)\mathrm{d}(\omega t);$$
$$b_k = \frac{1}{\pi}\int_0^{2\pi} f(A\sin(\omega t))\cos(k\omega t)\mathrm{d}(\omega t); \quad k = \overline{1,\infty}.$$

In this case we assume that in (4) $a_0 = 0$, in other words the constant component is absent.

The linear element is a low-pass filter, that is, when k is increasing the linear elements suppress higher harmonics. We will consider only the first harmonics. Then (4) will be presented in the form:

$$f(x) = a_1\sin(\omega t) + b_1\cos(\omega t), \tag{5}$$

where

$$a_1 = \frac{1}{\pi}\int_0^{2\pi} f(A\sin(\omega t))\sin(\omega t)\mathrm{d}(\omega t);$$
$$b_1 = \frac{1}{\pi}\int_0^{2\pi} f(A\sin(\omega t))\cos(\omega t)\mathrm{d}(\omega t).$$

From (3) you can write:

$$\begin{gathered}\sin(\omega t) = \frac{x}{A};\\ \cos(\omega t) = \frac{1}{A\omega}\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{1}{A\omega}\frac{\mathrm{d}}{\mathrm{d}t}x.\end{gathered} \tag{6}$$

Then we may rewrite (5) with respect (6):

$$f(x) = [\varkappa(A) + \frac{\varkappa'(A)}{\omega}\frac{\mathrm{d}}{\mathrm{d}t}]x = H_{nl}(A,\partial_t)x, \tag{7}$$

where $H_{nl}(A,\partial_t)$ — approximate transfer function of the nonlinear unit, $\varkappa(a)$ and $\varkappa'(a)$ are the harmonic linearization coefficients:

$$\begin{aligned}\varkappa(A) &= \frac{a_1}{A} = \frac{1}{A\pi}\int_0^{2\pi} f(A\sin(\omega t))\sin(\omega t)\mathrm{d}(\omega t);\\ \varkappa'(A) &= \frac{b_1}{A} = \frac{1}{A\pi}\int_0^{2\pi} f(A\sin(\omega t))\cos(\omega t)\mathrm{d}(\omega t).\end{aligned} \tag{8}$$

After finding the coefficients of harmonic linearization for given nonlinear unit, it is possible to study the parameters of the oscillation mode. The existence of oscillation mode in a nonlinear system corresponds to the determination of oscillating boundary of stability for the linearized system. Then A and ω can be found by using stability criteria of linear systems (Mikhailov, Nyquist–Mikhailov, Routh–Hurwitz). Thus, the study of self-oscillation parameters can be done by one of the methods of determining the limits of stability of linear systems.

3.1 The Nyquist–Mikhailov Criterion

This criterion belongs to analytical and graphic criteria. It has remarkable graphical representation of the system behavior and regions of existence of the oscillatory mode.

The Nyquist-Mikhailov criterion – [18] allows to judge about the stability of the open-loop automatic control system by using Nyquist plot (amplitude-phase characteristic) of the open-loop system.

Make the substitutions $\partial_t \to \mathrm{i}\omega$ and $s \to \partial_t \to \mathrm{i}\omega$ in the transfer function. Undamped sinusoidal oscillations with constant amplitude are determined by passing the amplitude-phase characteristics of the open-loop system through the point $(-1, \mathrm{i}0)$.

The characteristic function of the system is:

$$\begin{gathered} 1 + H_o(\mathrm{i}\omega) = 0, \\ H_o(\mathrm{i}\omega) := H_l(\mathrm{i}\omega) H_{nl}(A, \mathrm{i}\omega). \end{gathered}$$

where H_o — the transfer function of the open-loop system.

Thus:

$$H_l(\mathrm{i}\omega) H_{nl}(A, \mathrm{i}\omega) = -1. \tag{9}$$

Given by (7) from (9) the equality is obtained:

$$H_l(\mathrm{i}\omega) = -\frac{1}{\varkappa(A) + \mathrm{i}\varkappa'(A)}. \tag{10}$$

The left part of the Eq. (10) is the amplitude-phase characteristic of the linear unit, and the right part is the inverse of the amplitude-phase characteristic of the first harmonic non-linear level (with opposite sign). And the Eq. (10) is the equation of balance between the frequency and the amplitude.

This type of criterion is also called as a Goldfarb method.

Sometimes it is more convenient to write the Eq. (10) in the following form:

$$\varkappa(A) + \mathrm{i}\varkappa'(A) = -\frac{1}{H_l(\mathrm{i}\omega)}. \tag{11}$$

This type of criterion is also called as a Kochenburger method.

4 Harmonic Linearization of the Linearized RED Model

To rewrite the model (2) in the block-linear approach we need to linearize it. We will follow the article [6].

Let's write linearized system:

$$\begin{cases} \delta W(s) = -\frac{1}{s+\frac{N}{CT_f^2}(1+\mathrm{e}^{-sT_f})}\frac{C^2T_f}{2N^2}\mathrm{e}^{-sT_f}\delta p(s); \\ \delta Q(s) = \frac{1}{s+\frac{1}{T_f}}\frac{N}{T_f}\delta W(s); \\ \delta \hat{Q}(s) = \frac{1}{1+\frac{s}{w_q C}}\delta Q(s); \\ \delta p(s) = P_{\mathrm{RED}}\frac{1}{1+\frac{s}{w_q C}}\delta Q(s), \end{cases} \tag{12}$$

where the balance point is denoted by f index, variation is denoted by δ, and

$$P_{\mathrm{RED}} := \begin{cases} 0, & 0 < \hat{Q} \leqslant Q_{\min}, \\ \frac{p_{\max}}{Q_{\max}-Q_{\min}}, & Q_{\min} < \hat{Q} \leqslant Q_{\max}, \\ 0, & \hat{Q} > Q_{\max}. \end{cases}$$

Based on (12) the block representation of the linearized RED model (Fig. 3) is constructed.

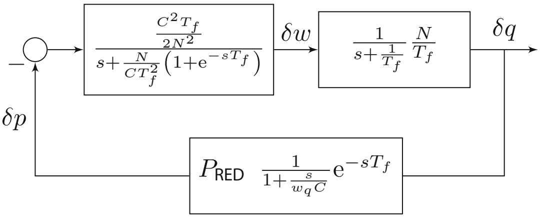

$\frac{\frac{C^2T_f}{2N^2}}{s+\frac{N}{CT_f^2}(1+\mathrm{e}^{-sT_f})}$
δw
$\frac{1}{s+\frac{1}{T_f}}\frac{N}{T_f}$
δq
$-$
δp
$P_{\mathrm{RED}} \quad \frac{1}{1+\frac{s}{w_q C}}\mathrm{e}^{-sT_f}$

Fig. 3. Block representation of the linearized RED model

Let's reduce the block diagram of linearized model (Fig. 3) to the form required for harmonic linearization.

As a static nonlinear function we will use P_{RED}. The linear part is follows:

$$\begin{aligned} H_l = {} & \frac{1}{s+\frac{N}{CT_f^2}(1+\mathrm{e}^{-sT_f})}\frac{C^2T_f}{2N^2}\mathrm{e}^{-sT_f} \times \frac{1}{s+\frac{1}{T_f}}\frac{N}{T_f} \times \frac{1}{1+\frac{s}{w_q C}} \\ & = \frac{1}{s+\frac{N}{CT_f^2}(1+\mathrm{e}^{-sT_f})}\frac{1}{s+\frac{1}{T_f}}\frac{1}{1+\frac{s}{w_q C}}\frac{C^2}{2N}\mathrm{e}^{-sT_f}. \end{aligned} \tag{13}$$

In the block representation the diagram from Fig. 3 will be as shown in Fig. 4.

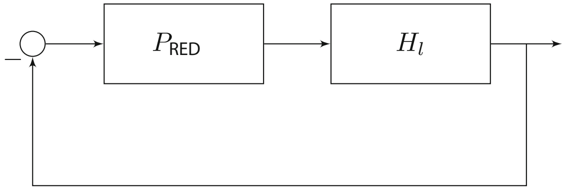


Fig. 4. Block representation of the linearized RED model for harmonic linearization

Let us compute the coefficients of harmonic linearization $\varkappa(a)$ and $\varkappa'(a)$ (8) for the static nonlinearity P_{RED}:

$$\varkappa(A) = \frac{4}{A\pi}\int_0^{\pi/2} P_{\mathrm{RED}}(A\sin(\omega t))\sin(\omega t)\mathrm{d}(\omega t);$$
$$\varkappa'(A) = \frac{4}{A\pi}\int_0^{\pi/2} P_{\mathrm{RED}}(A\sin(\omega t))\cos(\omega t)\mathrm{d}(\omega t).$$

We will get:

$$\begin{aligned}\varkappa(A) &= \frac{4}{A\pi}\frac{p_{\max}}{Q_{\max}-Q_{\min}}\int\limits_{\alpha_{\min}}^{\alpha_{\max}}\sin(\omega t)\mathrm{d}(\omega t)\\ &= \frac{4}{A\pi}\frac{p_{\max}}{Q_{\max}-Q_{\min}}-\cos(\omega t)\Big|_{\alpha_{\min}}^{\alpha_{\max}}\frac{4}{A\pi}\frac{p_{\max}(\cos\alpha_{\min}-\cos\alpha_{\max})}{Q_{\max}-Q_{\min}};\end{aligned} \tag{14}$$

$$\begin{aligned}\varkappa'(A) &= \frac{4}{A\pi}\frac{p_{\max}}{Q_{\max}-Q_{\min}}\int\limits_{\alpha_{\min}}^{\alpha_{\max}}\cos(\omega t)\mathrm{d}(\omega t)\\ &= \frac{4}{A\pi}\frac{p_{\max}}{Q_{\max}-Q_{\min}}\sin(\omega t)\Big|_{\alpha_{\min}}^{\alpha_{\max}} = \frac{4}{A\pi}\frac{p_{\max}(\sin\alpha_{\max}-\sin\alpha_{\min})}{Q_{\max}-Q_{\min}}.\end{aligned} \tag{15}$$

The values of sin and cos from integration limits $\alpha_{\min}$ and $\alpha_{\max}$:

$$\begin{aligned} x = A\sin\alpha_{\min} = Q_{\min},\quad \sin\alpha_{\min} = \frac{Q_{\min}}{A};\quad \cos\alpha_{\min} = \sqrt{1-\frac{Q_{\min}^2}{A^2}};\\ x = A\sin\alpha_{\max} = Q_{\max},\quad \sin\alpha_{\max} = \frac{Q_{\max}}{A};\quad \cos\alpha_{\max} = \sqrt{1-\frac{Q_{\max}^2}{A^2}}.\end{aligned} \tag{16}$$

Thus, from (14) and (15) with the help of (16) we will get:

$$\begin{aligned}\varkappa(A) &= \frac{4}{A\pi}\frac{p_{\max}}{Q_{\max}-Q_{\min}}(\sqrt{1-\frac{Q_{\min}^2}{A^2}}-\sqrt{1-\frac{Q_{\max}^2}{A^2}});\\ \varkappa'(A) &= \frac{4}{A\pi}\frac{p_{\max}}{Q_{\max}-Q_{\min}}\frac{Q_{\max}-Q_{\min}}{A} = \frac{4p_{\max}}{A^2\pi}.\end{aligned} \tag{17}$$

Thus, from (10), (13) and (17) we may derive:

$$\frac{1}{\mathrm{i}\omega + \frac{N}{CT_f^2}(1+\mathrm{e}^{-\mathrm{i}\omega T_f})} \frac{1}{\mathrm{i}\omega + \frac{1}{T_f}} \frac{1}{1+\frac{\mathrm{i}\omega}{w_q C}} \frac{C^2}{2N}\mathrm{e}^{-\mathrm{i}\omega T_f}$$
$$= -\frac{A\pi}{4p_{\max}}[\frac{1}{Q_{\max}-Q_{\min}}(\sqrt{1-\frac{Q_{\min}^2}{A^2}} - \sqrt{1-\frac{Q_{\max}^2}{A^2}}) + \mathrm{i}\frac{1}{A}]^{-1}. \qquad (18)$$

For clarity, it is possible to plot parametric graphs on the complex plane separately for left $H_l(\mathrm{i},\omega)$ and right $-1/H_{nl}(A)$ parts of the Eq. (18) (of ω and A respectively) (see Figs. 5 and 6). The intersection of the curves gives the point of emergence of self-oscillations.

For the example of the calculation we have chosen the following parameters: $Q_{\min} = 100$ [packets], $Q_{\max} = 150$ [packets], $p_{\max} = 0.1$, $T_p = 0.0075$ s, $w_q = 0.002$, $C = 2000$ [packets]/s, $N = 60$ (the number of TCP sessions). As a result we obtained the following values for the amplitude and the cyclic frequency: $A = 1.89$ [packets], $\omega = 16.55s^{-1}$.

The traffic behavior can be demonstrated by using the standard computer networks simulation software NS-2 [2,8]. For selected parameters we will get the

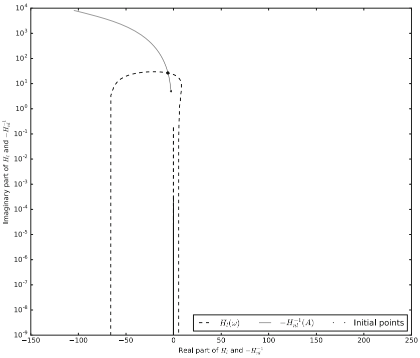


Fig. 5. Nyquist plot for system (18)

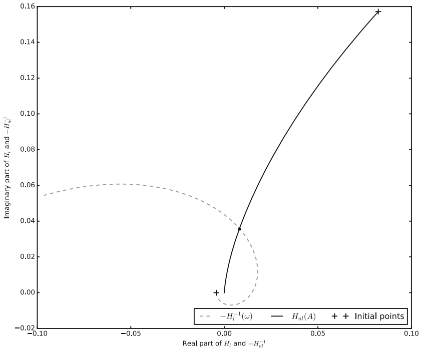


Fig. 6. Nyquist plot for system (11)

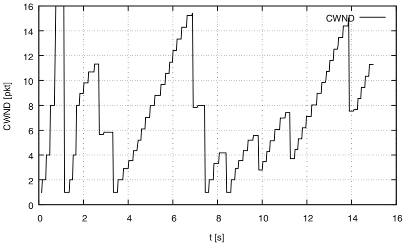


Fig. 7. A sliding window size changes at the source

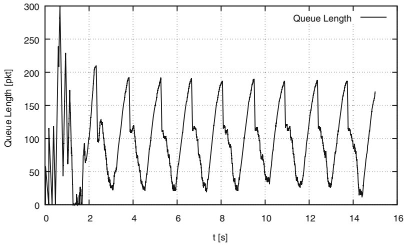


Fig. 8. A router's queue oscillation under RED control

graph of the window size change (at the traffic source) (Fig. 7) and oscillations of the instantaneous queue length at router under RED control (Fig. 8).

5 Conclusion

The authors demonstrated the technique of research of oscillatory modes of the systems with control. We tried to explain this technique for mathematicians unfamiliar with the formalism of the control theory. We plan to apply this technique to the study of a wide range of algorithms of traffic active control. Also it is interesting to compare these results with the previous results obtained for self-oscillation systems with control.

Acknowledgments. The work is partially supported by RFBR grants No's 15-07-08795 and 16-07-00556. Also the publication was financially supported by the Ministry of Education and Science of the Russian Federation (the Agreement No 02.A03.21.0008).

References

1. Adams, R.: Active queue management: a survey. IEEE Commun. Surv. Tutorials **15**(3), 1425–1476 (2013). doi:10.1109/SURV.2012.082212.00018
2. Altman, E., Jiménez, T.: NS simulator for beginners. Synth. Lect. Commun. Netw. **5**(1), 1–184 (2012). doi:10.2200/S00397ED1V01Y201112CNT010
3. Aström, K.J., Murray, R.M.: Feedback Systems: An Introduction for Scientists and Engineers, p. 408, USA (208)
4. Brockett, R.: Stochastic analysis for fluid queueing systems. In: Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304), vol. 3, pp. 3077–3082. IEEE (1999). doi:10.1109/CDC.1999.831407
5. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Trans. Netw. **1**(4), 397–413 (1993). doi:10.1109/90.251892
6. Hollot, C.V.V., Misra, V., Towsley, D.: A control theoretic analysis of RED. In: Proceedings of IEEE Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society, INFOCOM 2001 (Cat. No. 01CH37213), vol. 3, pp. 1510–1519. IEEE (2001). doi:10.1109/INFCOM.2001.916647
7. Hollot, C.V.V., Misra, V., Towsley, D.: On designing improved controllers for AQM routers supporting TCP flows. In: Proceedings of IEEE Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society, INFOCOM 2001, (Cat. No. 01CH37213), vol. 3, pp. 1726–1734. IEEE (2001). doi:10.1109/INFCOM.2001.916670
8. Issariyakul, T., Hossain, E.: Introduction to Network Simulator NS2. Springer, New York (2012). doi:10.1007/978-1-4614-1406-3. 9781461414
9. Jacobson, V.: Congestion avoidance and control. ACM SIGCOMM Comput. Commun. Rev. **18**(4), 314–329 (1988). doi:10.1145/52325.52356
10. Jenkins, A.: Self-oscillation. Phys. Rep. **525**(2), 167–222 (2013). doi:10.1016/j.physrep.2012.10.007

11. Korolkova, A.V., Kulyabov, D.S., Chernoivanov, A.I.: On the classification of RED algorithms. Bull. Peoples' Friendship Univ. Russ. Ser. “Math. Inf. Sci. Phys.” (3), 34–46 (2009)
12. Korolkova, A.V., Kulyabov, D.S., Sevastianov, L.A.: Combinatorial and operator approaches to RED modeling. Math. Model. Geom. **3**(3), 1–18 (2015)
13. Korolkova, A.V., Velieva, T.R., Abaev, P.A., Sevastianov, L.A., Kulyabov, D.S.: Hybrid simulation of active traffic management. In: Proceedings 30th European Conference on Modelling and Simulation, pp. 685–691 (2016). doi:10.7148/2016-0685
14. Kushwaha, V., Gupta, R.: Congestion control for high-speed wired network: a systematic literature review. J. Netw. Comput. Appl. **45**, 62–78 (2014). doi:10.1016/j.jnca.2014.07.005
15. Lautenschlaeger, W., Francini, A.: Global synchronization protection for bandwidth sharing TCP flows in high-speed links. In: Proceedings of 16th International Conference on High Performance Switching and Routing, IEEE HPSR 2015, Budapest, Hungary (2015)
16. Misra, V., Gong, W.B., Towsley, D.: Stochastic differential equation modeling and analysis of TCP-windowsize behavior. In: Proceedings of Performance 1999 (1999)
17. Misra, V., Gong, W.B., Towsley, D.: Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. ACM SIGCOMM Comput. Commun. Rev. **30**(4), 151–160 (2000). doi:10.1145/347057.347421
18. Nyquist, H.: Regeneration theory. Bell Syst. Techn. J. **11**(1), 126–147 (1932). doi:10.1002/j.1538-7305.1932.tb02344.x
19. Ren, F., Lin, C., Wei, B.: A nonlinear control theoretic analysis to TCP-RED system. Comput. Netw. **49**(4), 580–592 (2005). doi:10.1016/j.comnet.2005.01.016
20. Velieva, T.R., Korolkova, A.V., Kulyabov, D.S.: Designing installations for verification of the model of active queue management discipline RED in the GNS3. In: 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp. 570–577. IEEE Computer Society (2015). doi:10.1109/ICUMT.2014.7002164

Effectiveness Examination of a Multi-channel CSMA/CA Detector

Dariusz Laskowski[1(✉)], Marcin Pólkowski[2], Piotr Łubkowski[1],
and Leszek Nowosielski[1]

[1] Military University of Technology, Gen. S. Kaliskiego 2 Street,
00-908 Warsaw, Poland
{dariusz.laskowski,piotr.lubkowski,
leszek.nowosielski}@wat.edu.pl
[2] Transbit Sp. z o. o., Łukasza Drewny 80 Street, 02-968 Warsaw, Poland
marcin.polkowski@transbit.com.pl

Abstract. Along with the continuous development of wireless networks a problem with the availability of free bandwidth occurs more often. The amount of distortion is growing at an alarming rate, which cannot be containing in any way. The only solution is to develop systems that can dynamically adjust to the prevailing conditions. The article presents a description of the tests performed CSMA/CA (EMCD-Effective Multichannel Detector) multichannel detector designed by Transbit Sp. z o. o. in the framework of the project financed by Polish National Centre of Research and Development for advanced spectrum management. Testing the detector has been implemented in the MATLAB simulation environment.

Keywords: OFDM · CSMA/CA · Multi-channel

1 Introduction

Most modern wireless networks IP (Internet Protocol) based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) access method. It is different from that used in wired Ethernet networks, in which collisions are detected and frame retransmissions are enforced. The main feature of the CSMA/CA is to avoid the emergence of conflicts on the basis of the listening network status. If two stations at the same time want to send data, the only one of them will be able to do it. CSMA /CA properties are the object of many researches i.e.: analysis of priority arbitration with throughput optimization and prediction [1, 2], optimization in heterogeneous networks [3] or random CSMA networks [4]. In the study of new network solutions CSMA should perform a test. The test requires reliable equipment. Therefore, the authors decided to develop and implement a CSMA multichannel detector. The article presents a description of the tests performed multichannel detector CSMA/CA taking into account the legal nature of the project. Small networks with a small amount of the station seem to work well. In the case of a complicated topology, the waiting time for giving data may be too long from the point of view of user's expectations. Therefore, the idea of action network on several frequency channels at the same time forced a

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_22

quick suitable detector which detects signals carrier on all channels. Testing and the results described in the article relate to the detector as described in [5, 6].

2 Testbed

The tested detector has been implemented in simulation environment Matlab. The detector itself consists of nine modules supported by the main program (detector). Additional modules simulate the phenomenon of interference. The most important modules are described in the following next subsections.

2.1 Pseudorandom Numbers Generator

A pseudorandom generator creates a batch data for the construction of the OFDM signal. The diagram shows the following Fig. 1.

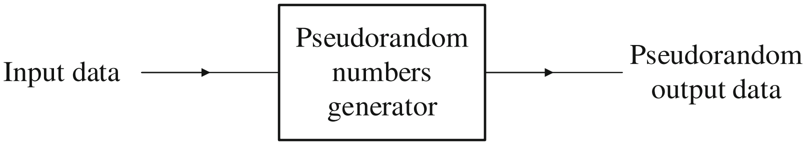


Fig. 1. Pseudorandom numbers generator scheme

2.2 Inverse Fast Fourier Transform Module (IFFT)

The data of the generator are used to build the preamble and OFDM symbols [7, 8]. Subcarriers of this modulation are modulated using QPSK modulation. Framer is responsible for correct formation of OFDM symbols. Figures 2 and 3 show respectively the OFDM signal generated without interference, and its spectrum.

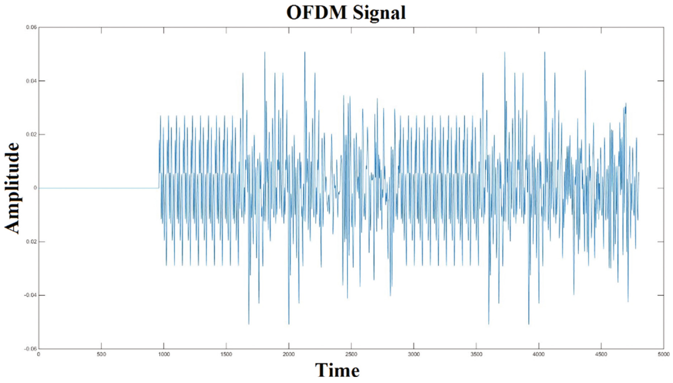


Fig. 2. Generated OFDM signal

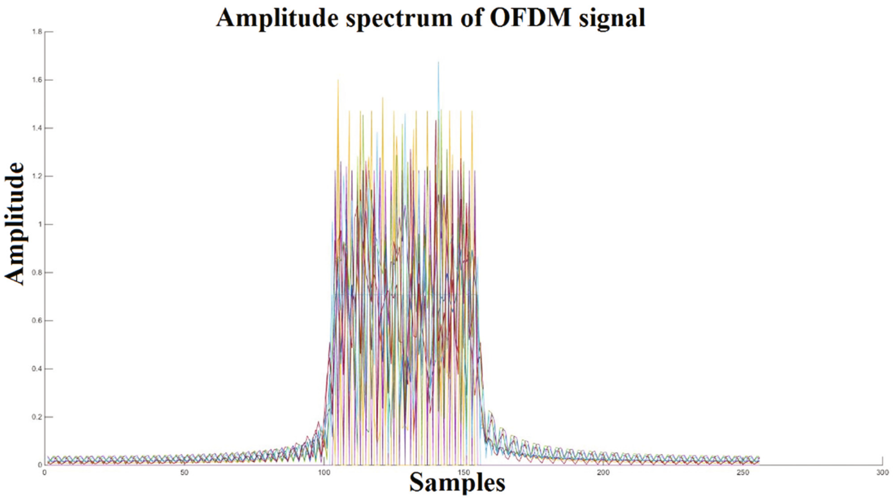


Fig. 3. Amplitude spectrum of generated signal

2.3 Additive White Gaussian Noise Module (AWGN)

Generated OFDM signal is transmitted by the noised channel. It was created by using the White Noise Generator with Gauss distribution (AWGN). As an input to this function takes the OFDM signal, while the regulated parameter is the target signal-to-noise ratio (SNR) in decibels (Fig. 4).

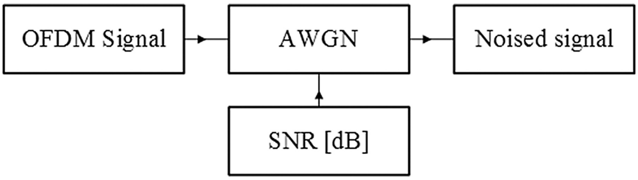


Fig. 4. AWGN module scheme

In addition, to check the resistance of the quality detection in the Simulator, AWGN function at the input is the sum of the OFDM signal and narrow-band interference.

2.4 Low-Pass Filter

Distorted (degraded) signal by AWGN is given at the input of a bandpass filter matched to the width of the currently used channel. Because the spectrum of the signal consisted of two symmetrical halves received as a result of digital Fourier Transforms are

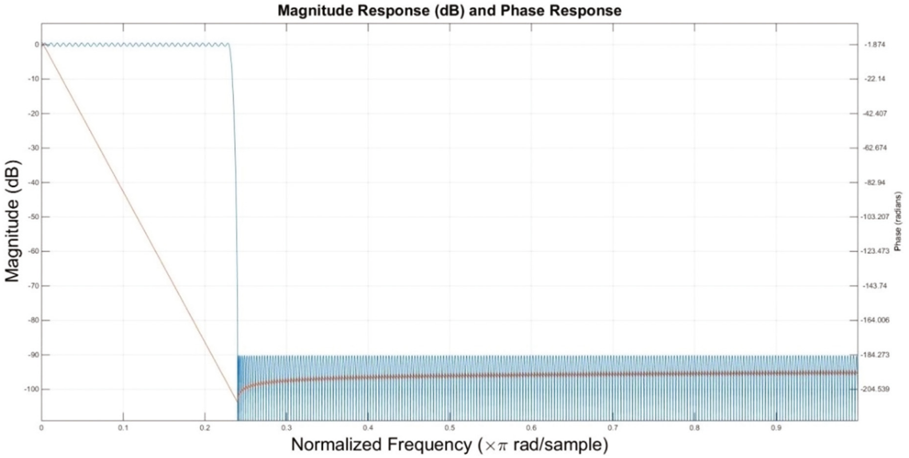


Fig. 5. Low-pass filter characteristic

designed one low-pass filter (Fig. 5). After filtering the halves of the spectra have been made in the complete spectrum of the signal using the Matlab fftshift (moves the sample spectrum, so that a sample of zero was in the middle of the graph).

2.5 Energy Detector

Degraded signal, after filtering, is scanned in the channel by using energy detector, which detects a signal level exceeding the set threshold, which triggers the correlative detector (Fig. 6). Energy detector performs a summation of the squared values of samples of OFDM signal in the length of 320 samples. This window corresponds to the length of one OFDM symbol. After the calculation of the energy value for the duration of one window, it compares its level from a previously defined threshold. The threshold was schedule on the basis of receivers parameters (self-noise and the width of the channel).

Achieving the required threshold of detection triggers correlative detector.

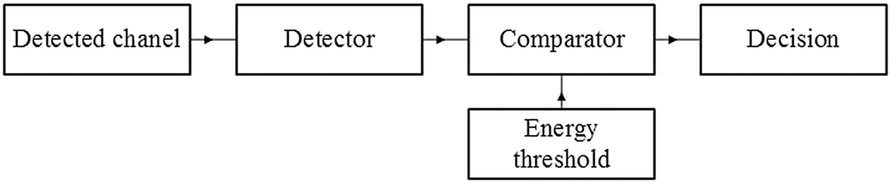


Fig. 6. Energy detector module scheme

2.6 Correlative Detector

The layout of the correlative detector is based on a method of Schmidl & Cox ‘ 97. Degraded OFDM signal is given to entry of the detector. Burst and peak detectors are closely related. The first detects the signal in the window that has been declared in the transmitter. In this window the peak detector, based on the analysis of these dwellings, generates a correlation peaks and counts them. If it detected all peaks (10 on the symbol), the correlative detector tool would tell you that the signal has correct synchronization (Fig. 7).

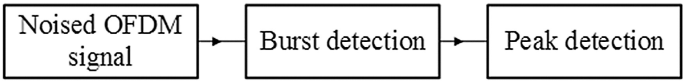


Fig. 7. Correlative detector module scheme

3 Tests

3.1 Tests Description

The test scenarios consist in validation the detection process determined by the probability of correct detection (P_{ps}) in the channel with white noise and narrowband jamming signal. The study was performed for two channel width: 1 and 2 MHz. The ratio of the amplitude of the interference to OFDM signal is given in decibels:

$$\frac{x_{dist}}{x_{OFDM}}[dB] = 20log_{10}\frac{x_{dist}}{x_{OFDM}} \tag{1}$$

where:
X_{dist} – jamming signal amplitude,
X_{OFDM} – OFDM signal maximal amplitude.

The following tests have been performed for each channel: at a constant signal-to-noise ratio changed the level of interference in OFDM signal levels in the range of −40 to 0 dB. The measurements were repeated 1000 times to determine the probability value of correct synchronization. The above study was repeated for SNR = 6, 10, 15, 20, 25, 30 dB. SNR = 6 dB noise value of their own real receiver.

3.2 Results

The chart depicted in Fig. 8 illustrates the OFDM signal in time domain, energy level of the signal in the receiver, and the moment of reaching the energy threshold level that was defined in the detector. The point at which the detector triggered the process of correlative detection was delayed by reaching a threshold due to the measurement of energy in the length of 320 samples. Once it has registered in its power window exceeding the threshold detector starts correlative algorithm.

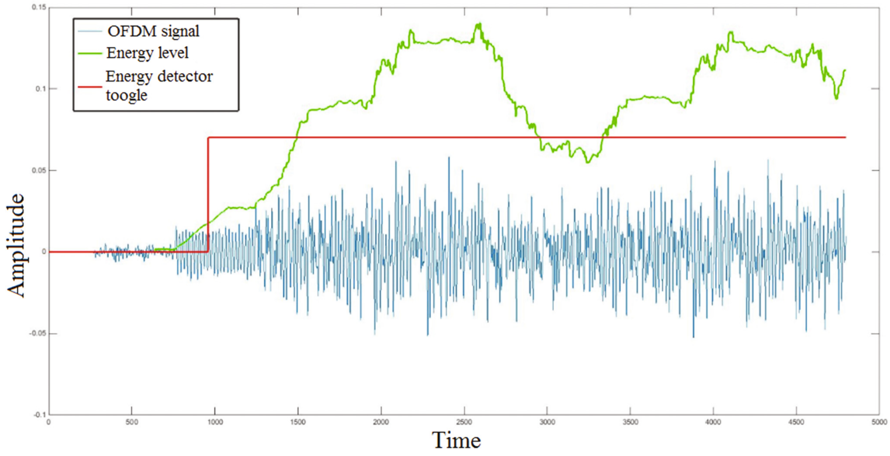


Fig. 8. Noised OFDM signal

In the case of the channel with white noise without interferences, work of correlative algorithm was not disrupted in any way. Because the generated noise has a flat frequency characteristic, did not affect significantly the outcome of cross-analysis.

Figure 9 shows the measurement results for OFDM signal in a channel with a width of 1 MHz. For SNR equal 6 and 10 dB were obtained the worst results.

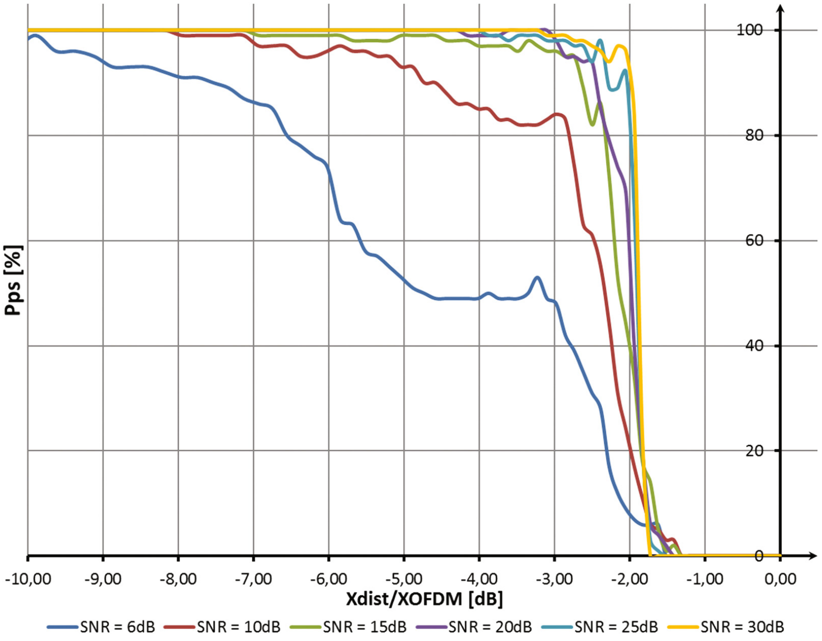


Fig. 9. Probability of correct detection on 1 MHz channel

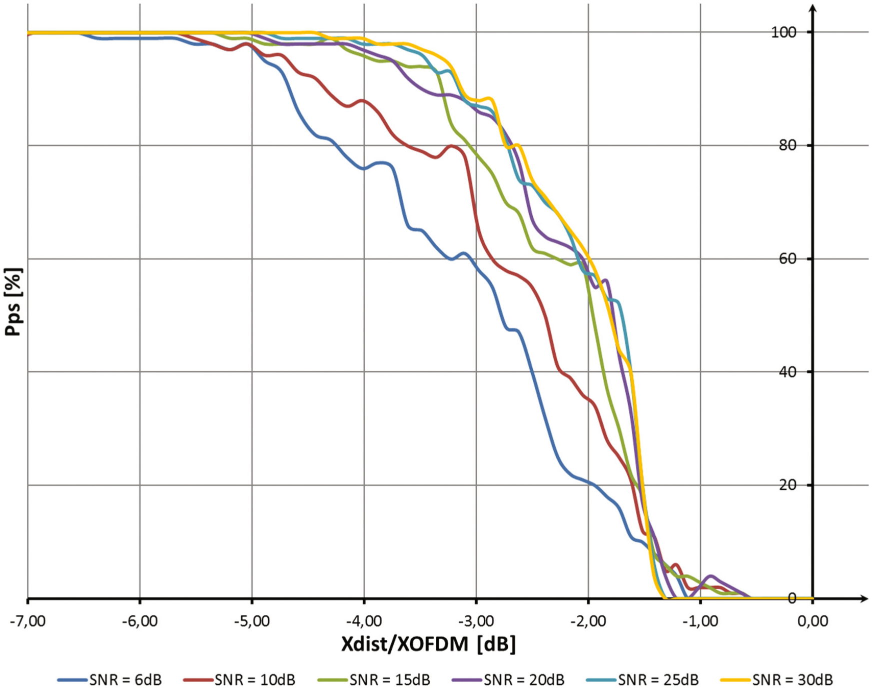


Fig. 10. Probability of correct detection on 2 MHz channel

Above these values narrowband interfering signal amplitude equal to half of the OFDM signal amplitude (−3 dB) does not interfere with the work of the correlative detector and the probability of correct detection was oscillating within the limits of 95–100%. By SNR = 6 dB, 100% probability of correct detection was for $x_{dist}/x_{OFDM} \leq 10$ dB. For the channel with a width of 2 MHz acceptable probability value occurred for relative $x_{dist}/x_{OFDM} = -4$ DB (at SNR > 10 dB). This channel has proven to be more resistant than 1 MHz channel noise with high amplitude (Pps = 100% for $x_{dist}/x_{OFDM} = -6$ dB at SNR = 6 dB, Fig. 10).

Studies have shown that a low signal-to-noise ratio (0 dB), in the absence of narrow-band interference, does not adversely affect the process of detection. With narrow-band interference, the channel on a narrower band showed lower effectiveness of correct detection than the channel with a wider band.

4 Conclusions

Continuous increase in the use of wireless telecommunications networks forced work on methods of a dynamic and effective management of available frequency band. The application of multi-channel access methods allows for a significant increase in network capacity and its resistance to interference. Research has shown that by increasing the width of the channel increasing the bandwidth of the channel as well as its resistance to distortion of narrowband. Thanks to increasing effective throughput to useful data.

Detector described in the article served as the research and development project for create multichannel broadband radio by Transbit Sp. z o. o. Has it up to 4 MB/s in one channel and range up to 30 km, while maintaining the quality of the transmission even in a mobile application. Next, it is intended to extend research on the content presented in the articles.

References

1. Ashrafuzzaman, K., Fapojuwo, A.O.: Analysis of priority arbitration in low-rate CSMA/CA-based differentiated access with throughput optimization. Int. J. Commun. Syst. **30**(1) (2017). ISSN 1074-5351. doi:10.1002/dac.2922
2. Hamamoto, R., Takano, C., Obata, H., Ishida, K.: Improvement of throughput prediction scheme considering terminal distribution in multi-rate WLAN considering both CSMA/CA and frame collision. IEICE Trans. Inf. Syst. **E99D**(12), 2923–2933 (2016). doi:10.1587/transinf.2016PAP0019
3. Andrews, M., Zhang, L.S.: Utility optimization in heterogeneous networks via CSMA-based algorithms. Wirel. Netw. **23**(1), 219–232 (2017). doi:10.1007/s11276-015-1149-z
4. Bienenstock, A., Bergel, I.: The performance of random CSMA networks with threshold scheduling. Trans. Emerg. Telecommun. Technol. **27**(11), 1550–1562 (2016). doi:10.1002/ett.3096
5. Lubkowski, P., Laskowski, D.: Test of the multimedia services implementation in information and communication networks. In: Advances in Intelligent Systems and Computing. Springer, Switzerland, vol. 286, pp. 325–332 (2014). ISSN 2194-5357, ISBN 978-3-319-07012-4 (Print). doi:10.1007/978-3-319-07013-1_31
6. Laskowski, D., et al.: The concept of the effective multi-channel CSMA/CA detector. In: Advances in Intelligent Systems and Computing. Springer, Switzerland, vol. 470, pp. 323–331 (2016). ISSN 21945357, ISBN 978-331939638-5. doi:10.1007/978-3-319-39639-2_28
7. Malon, K., Lopatka, J.: Efficient methods of interference suppression for spectrum pooling cognitive radios using OFDM signals. In: 5th International Conference on Signal Processing and Communication Systems (2011). ISBN 978-1-4577-1180-0
8. Malon, K., Lopatka, J.: Spectrum sharing of OFDM signals for cognitive radios. In: 4th International Conference on Signal Processing and Communication Systems (2010). ISBN 978-1-4244-7907-8

IaaS vs. Traditional Hosting for Web Applications - Cost Effectiveness Analysis for a Local Market

Paweł Lorenc and Marek Woda(✉)

Department of Computer Engineering, Wroclaw University of Technology,
Janiszewskiego 11-17, 50-372 Wrocław, Poland
pawlakzn@gmail.com, marek.woda@pwr.edu.pl

Abstract. This work was carried out in order to examine the legitimacy of use and to compare ongoing cost of cloud based (IaaS) and classic hosting for three types of web applications. With a plethora of various cloud providers, increased reliability and plummeting down daily ongoing cost – Infrastructure as a Service seems to be a natural successor of traditional hosting for even fairly simple web applications. However, there was unclear until now, how IaaS compares to a traditional hosting in terms of cost-effectiveness. In that article three different types of sites were used to provide a clear answer which type of hosting to use to justify ongoing cost and the same time assure an optimal performance of a certain types of web applications.

Keywords: Cloud hosing · Traditional hosting · Web applications · Resources consumption · Cost effectiveness analysis

1 Introduction

In the contemporary world, Internet is an indispensable element for most companies in all significant market areas; a basis for the functioning of majority of the government agencies, NGOs and large portion of individuals. Statistics [10, 11, 19, 21] indicate that approx. 50% of world's population has access to Internet, there is around about 1,8 billion active webpages, 6+ million computers online 24/7 and connected to the global network, and lastly 42 PT (!) bytes of data transferred every second. Internet has become a vital aspect for an increasing number (as for now 6,4% of total employed people) of professionally active people that work from home on a daily basis. Nowadays, it is not a difficult task to create even a complex website - there are many tools [12, 13] that isolate a user from the technical side - so that he could focus only on their needs, that is, filling the created site content. These solutions offer a range of capabilities desired by surfers, and everything is delivered in a very user friendly way. Every newly created web page or application needs a server to run on. Nowadays, there are two practical options: traditional hosting (on a dedicated server) or hosting in a cloud. Choosing the appropriate type of hosting for a web application may be difficult.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_23

2 To Cloud or not

In times of shrinking budgets many companies [5, 14], from small and medium-sized businesses are looking for new ways to effectively ensure their web hosting needs. Hosting environment is rapidly changing, and many people are now looking beyond traditional hosting towards cloud one. Traditional approach comes mainly in two forms, dedicated and shared. With dedicated hosting, a company pays for the complete resources of one or more servers from a service provider. The client has full control over a fixed amount of resources (dedicated bandwidth, CPU, RAM, and drive space). With shared hosting, which is more common among small and medium sized businesses, the client pays for a set amount of space (storage) on a single server, and that server's resources are shared by a number of other websites. Traditional hosting has drawbacks [2, 3]. Because the resources of a single server are shared among a number of different websites, spikes in traffic to those websites can mean decreased performance for your own. Security breaches and other performance issues on other sites make take yours down as well. And there's a single point of failure. If the server itself experiences technical problems, everyone hosted on that server will be affected. Cloud hosting offers a level of scalability that traditional hosting can't. Instead of paying for a set amount of space upfront on a single server, the user pays as they go for what they actually use. With cloud hosting, the load is balanced across a cluster of multiple servers. The information and applications contained on those servers are mirrored across the whole cluster, meaning that if an individual server goes down, there is no lost information or downtime. Because of this redundancy, cloud hosting is much more elastic and resilient. Problems with one website or application are unlikely to affect your bandwidth or performance. Cloud hosting companies provide Infrastructure-as-a-Service (IaaS) [18], which is a s-tandardized, highly automated offering, where compute resources, complemented by storage and networking capabilities are owned and hosted by a service provider and offered to customers on-demand. Customers are able to self-provision this infrastructure, using a Web-based graphical user interface that serves as an IT operations management console for the overall environment. IT departments needn't to invest in in-house server hardware. And customers don't need to pay for up front for extra storage or processing capacity that they don't use. Cloud hosting is more quickly scalable than traditional hosting [9]. If an application or website receives more or less traffic, the cloud servers scale up and down automatically. With cloud hosting, there's no need to manually add or remove server space as there is in shared hosting. Cloud hosting is still a relatively new technology, and many who have experience with traditional hosting are reluctant to move to something different. Finding some suitable criteria for comparison IaaS and traditional hosting is a major issue. It was decided that besides technical parameters, a key constraint influencing the decision are ongoing monthly costs. Some studies have shown [15–17, 20] that a cheaper and more efficient may be the cloud. Cloud allows usually to handle a larger number of users at lower cost, which is a plus. However, one should pay attention to the characteristics of website traffic and the budget that was intended to allocate and obviously local market conditions. The research was conducted for three, diverse types of websites. The results will give a clear answer what kind of hosting is more suitable for which type of web applications on Polish market in late 2016.

3 Factors Influencing the Decision About Hosting

The final choice on hosting type is due to the different priorities and frequently a matter of a subjective decision. Given below factors helped us to structure research and make fundamental decisions.

3.1 Hidden Costs

Common sense dictates that the costs associated with a certain hosting solution is the most important aspect upon decision making. Everyone wants to spend as little as possible, and at the same time to achieve the best results. However, it is essential to address following question: How much capital do we need to run our website efficiently (performance wise) and cost-effectively? One should answer this question in the perspective of different time periods: one month, six months, year, or even two or three years. Aside from a hosting offer, other a (local) factor is heavily influencing the final cost – it is the exchange rate. While the dedicated servers have no problems with the settlement in any currency (almost every country has local products), whereas, in the case of cloud solutions most of them charge only in USD or EUR. Unfortunately, tested by us, Amazon cloud does not allow settling in Polish zlotys. These issues lead us the next uncertainty: the ability to predict costs. Hosting market is changing rapidly which leads to difficulties in predictability of costs in longer periods of time.

3.2 Expertise

Building a website and decision where to host it, are two essentially different services. Making a decision on the type of hosting still requires the physical implementation. At the same time, it is not a one-time task. What is needed is an experienced administrator with a certain knowledge: about the operating systems (usually Linux) and technologies used by the hosting site. In the cloud, one knows the architecture and the options available in the selected service provider. With it, one can efficiently and cheaper set up the server. With a traditional web hosting often happens that the administrator himself must take care of virtual machines on a separate *Virtual Private Server* and creation of backups, which are easily available in the cloud. In addition, in the cloud it is more difficult to configure a particular (optimal for our needs) instance. This is due to the fact that it can consist of many components: a separate service for file storage, database management or additional drives.

3.3 Prospect of Possible Changes

A lot may depend on the period that we have to analyze. If it is longer (over a year), an additional factor is the possibility of future site customizations. It may be due to technical issues and/or popularity of the page. Technical changes can lead to change in server requirements, since a website/application may require more resources over time.

This in turn forces the analysis of how difficult it will be to change the current version of the hosting to another (usually more expensive and time consuming). Much easier is to predict whether in given time frame there will significant technical changes occur, rather than predict its near time statistics (number of visits, load). Application owners that count on their website high popularity must be prepared for consequences related to it. For this reason, it carried out various tests and generates appropriate load estimate. The more agents, the more difficult their effectiveness.

3.4 Risks

One of risk that should be mitigated is the potential unavailability of a site due to a failure of various nature (e.g. server failure, high load with limited resources, power outage etc.). It is common knowledge that uptime of the cloud servers is typically higher. In the case of physical damage of a dedicated server it may not be available for a longer while. The cloud hosting through replication avoids this problem. It is imperative to pay attention to the volatility of the cost. In the case of the dedicated hosting, cost for a defined period of time is fixed and known from the beginning. IaaS price is variable, calculated based on pay-per-use scheme. Some sudden spikes of load or unexpected influx of new users may result in unplanned, excessive costs. However, there are methods to avoid such a scenario (notifications and locks).

3.5 Anticipated Results

Review of available hosting offers (of dedicated servers and IaaS) does not imply an easy way how to compare them. Whereas once we can grasp the final price of traditional hosting solutions, TCO of IaaS offering without a detailed analysis, remains mystery. There are a few, rather superficial and generic research made [4, 6–8, 16, 17] that made attempt to address the question which hosting solution to choose. Unfortunately, none has addressed significant nuisances like different type of web application, different type of server load or local market conditions. Authors' assumption was, both approaches have to be competitive. If it were not so, one of them would not exist or would significantly lost its presence on the market.

Since there is no "common knowledge" which is better in terms of monetary resources, it is assumed that the cost-effectiveness of hosting services on a dedicated server or cloud will "overleap" each other like on Fig. 1. Diagram in Fig. 1. depicts authors assumptions, how the final results may look like. As one can see, the graph for the classical hosting is non-linear (stepwise). One pays for the current hosting option until it turns out to be inadequate to the load and one switches to a more expensive one. Cloud chart is unpredictable, non-linear and with many variations. The costs vary in a short period of time. The greatest riddle was the difference between these charts in real world conditions. It was only assumed that for a certain load a better choice may be traditional web hosting, and for another cloud based one.

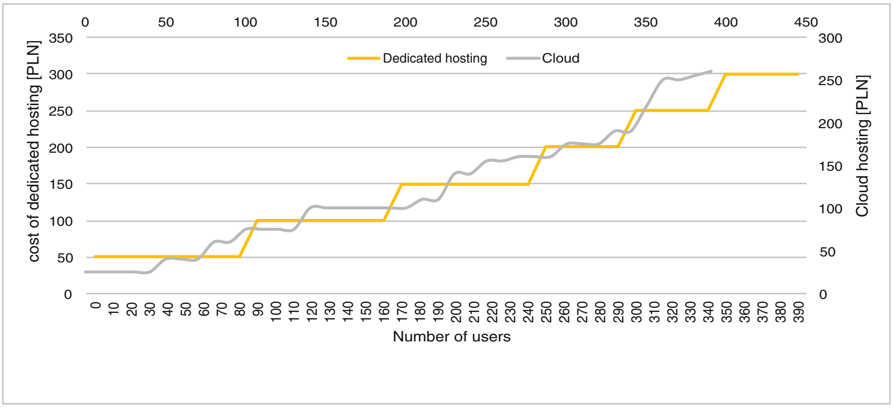


Fig. 1. Assumed cost of hosting (dedicated vs. cloud based) for a various number of users

4 Test Environment

As a basis for experiments we developed three types of websites (a *brochure*, *standard* and *business* one) that were later hosted on a dedicated server (provided by OVH – best value in the market) and in Amazon Cloud, which was chosen due its undisputed popularity.

Table 1. Technical parameters of a dedicated server used for tests

Dedicated server type	E5-SAT-2-32
CPU	Intel Xeon E5 1650 3.1 GHz + (6 cores/12 threads)
RAM	32 GB DDR3
HDD	2 × 3 TB SATA
RAID	Soft
Network link	1 GBit
Bandwidth	250 Mbps
Transfer	unlimited
IPv4/IPv6	1/64

A *brochure* was a very simple website (purely HTML based), which was mainly used to display information. It didn’t require a database. It consisted of up to a few pages (in our case it was a mere 2 MB total disk space). As for a *standard* site, we considered an application (with a database) that interacted with users. It was CMS based (WordPress) webpage of popular chain of restaurants (in our example: dB size 800 MB+, project files around 2 GB). As for *business* site, we considered a large set pages and an engine that required considerable amount of computations and intense database operations (it was a large Web-shop, based on *Magento* engine). The test environment was prepared to accommodate requirements of test websites, so both standard (based on WordPress) and business application (Magento) require PHP along with a database (MySQL). All websites were run on Ubuntu 14.04 with Apache 2.4 as a web server.

4.1 Dedicated Server – Traditional Hosting

Thanks to free software called *Proxmox* we were able to utilize resources of the dedicated server (Table 1.) as six Virtual Private Servers (Table 2). Monthly cost of the dedicated server was (304,93 PLN gross) which was rounded down to 300 PLN.

Table 2. Technical parameters of 6 VPS - derived from the dedicated server.

Name of VPS	Monthly fee [PLN]	# of cores	RAM [MB]
Ubuntu1	50	1	5461
Ubuntu2	100	2	10923
Ubuntu3	150	3	16384
Ubuntu4	200	4	21845
Ubuntu5	250	5	27307
Ubuntu6	300	6	32768

4.2 Cloud Hosting - IaaS

As a cloud service provider, Amazon Web Services (AWS) was selected. This is one of the largest companies providing such services. An additional argument to use AWS was able to benefit from free testing. This offer is available for a year after opening an account. And some services (including ones we used), for some tenant instances are free for a period of 750 h of operation. It was very helpful during the initial server configuration. The basic service used during the tests was Elastic Compute Cloud (EC2). Thanks to it one can create instances of the web servers. AWS offers 40 different configurations, which differ in the number of vCPUs, memory, disk type, storage size and a price. The basic unit of account is the price per hour. Significant reductions in price can be achieved by purchasing services for a year or three years in advance (Table 3).

Table 3. Price (per hour of usage) of selected (EC2 instances EU- Frankfurt).

Instance	vCPU	RAM (GB)	Storage (GB)	Price (USD/h)
t2.nano	1	0,5	EBS	0,0075
t2.micro	1	1	EBS	0,0150
t2.small	1	2	EBS	0,0300
t2.medium	2	4	EBS	0,0600
t2.large	2	8	EBS	0,1200

It was decided to test our test websites in five instances, which were available in competitive prices in relation to the dedicated server. EBS is an abbreviation of Elastic Block Store, and acts as a disk storage (SSD based) that stores files for EC2 instances (self-replicable, highly available and durable). In order to fully utilize the potential of the cloud for database operations RDS service was used. In the case of RDS prices vary due to the type of database (in our case MySQL) and a region (in our case EU – Frankfurt was selected) (Table 4).

Table 4. The parameters and the price of RDS instances (MySQL in EU- Frankfurt).

Instance	vCPU	RAM (GB)	Price (USD/h)
db.t2.micro	t	1	0,020
db.t2.small	1	2	0,040
db.t2.medium	2	4	0,080

4.3 Tests and Measurements

The complete test scenario for a *brochure*, consisted in displaying the contents of all subpages. For a *standard* page, following order of actions was considered as a complete test scenario: opening main page, go to list of restaurants, choose one, choose menu, list of menu entries, pick an entry (open PDF), go to page with categories of recipes, choose a recipe, open contact page, fill and send a contact form. For a *business* page a test scenario consisted in: opening main page, go to one of static pages, choose a category of products, choose a product and its size, add a product to the shopping cart, go to shopping cart page, choose a buy as a guest option, fill in delivery address, go to the payment page, choose the form of payment, confirm the purchase and submit the form. All the measurements during test scenarios were taken/processed by following tools: *Gatling*, *collectd*, *Graphite*. During the tests following parameters were collected [1]: ***total # of simultaneous requests, total test time*** (times of user's initialization and a test scenario preparation was included), ***time of users' initialization*** (included in the ***test time***, but defined separately), ***% of fulfilled requests*** *(measured in a given time-frame,* which could fall into one of following predefined ranges (in ms): *>800, 800-1200, <1200 and failed* (> 60000)). If % of unfulfilled requests was higher than 1% it was considered as failed, *%* ***CPU/RAM Usage***, **Server Load** (for last minute), ***# of requests per second***. In case of IaaS, ***CPU*** and ***RAM usage*** was additionally measured on a database instance.

5 Analysis of Results

One of the most important selection criteria is cost. And it was chosen as a parameter that can be used for comparison of the two hosting types. Cost of hosting on a dedicated server was trivial to define, since it was a simple monthly fee. Whereas the total cost of cloud based hosting (AWS), was far more intricate to project and influenced by the following factors: choice of EC2 and RDS instance type, storage type and its size, dB size, time of use. In our case, data was stored only in EC2 and not in S3 so not additional cost for data transfer was incurred. The initial examination of AWS offer, proved to be excessively expensive and simply unreasonable in comparison to offerings of traditional hosting from the local (Polish) market. In order to make it more money-wise, it was decided to use two different instances - one with better parameters (more capable) for 18 h. a day (where the load would be potentially higher) and other during remaining 6 h. (at night) the less capable one. Amazon is very flexible in terms of form of charging. Pay per use is the most expensive, whereas for payment in advance one may get accordingly 20–34% (depending on selection of a certain instance) discount for 1 year and 46–55% for 3 yrs. Taking advantage of promotional pricing, requires allocation of large portion of a budget before the application becomes profitable, which may not be suitable for everyone. At the same time, it collides with idea of flexible solution, which does not require large budget allocation for long period of time (Table 5).

Table 5. Max. number of fulfilled requests – IaaS vs. Traditional hosting

VPS	Brochure		Standard		Business		*IaaS*
Ubuntu1	680	680	280	50	140	20	t2.nano
Ubuntu2	680	680	360	100	280	45	t2.micro
Ubuntu3	680	680	410	120	330	70	t2.small
Ubuntu4	680	680	420	190	370	110	t2.medium
Ubuntu5	680	680	420	190	390	120	t2.large
Ubuntu6	680	x	420	x	410		

Performance tests revealed that for *Standard* and *Business* websites, a single VPS (the least capable) is able to meet almost twice higher load (in terms fulfilled requests than the most expensive (most capable) variant of cloud instances. Due its simplicity *brochure* website was independent of hardware - no matter of any variant of cloud or traditional hosting was used. The bottleneck constituted here rather a webserver than any other technical aspect. *Brochure* was the only case in the comparison where its hosting was more affordable in the cloud regardless of a hosting time-frame (monthly – 23 vs. 50 PLN and annually – 189 vs. 600 PLN) (Fig. 2).

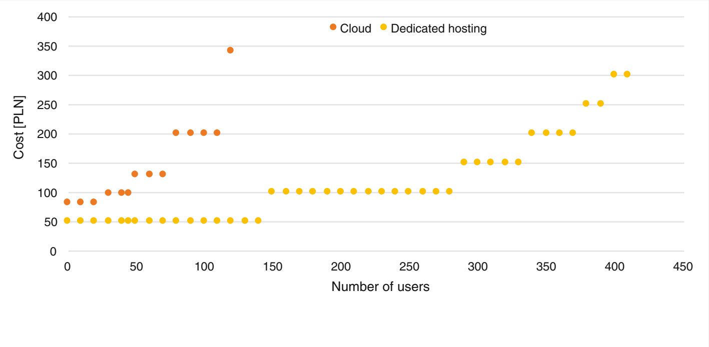


Fig. 2. *Business* website - monthly cost of hosting for max # of user

For both the *standard* and *business* sites, cloud hosting turned out to be more expensive and less (capacity) efficient solution. Results for traditional hosting demonstrated that another (more expensive) VPS instance proved to be sufficient for a larger number of users. In the case of the *Business* website, the first instance (Ubuntu1) of the server allowed to handle 140 requests, another 280 (twice more), following next one 330 and the last one 370. For cloud hosting, the transition to more efficient configuration can mean considerable sudden increase in costs (e.g. from 200 to 340 PLN). One does not get significant increase of capacity but cost soars irresponsibly (Fig. 3).

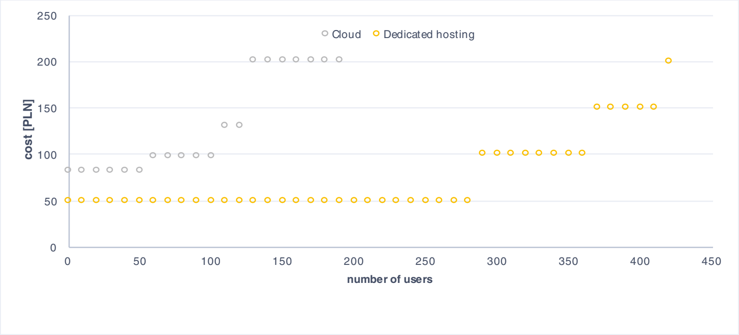


Fig. 3. *Standard* website - monthly cost of hosting for max # of user

6 Conclusions

The work presented in the paper was devoted to the study cost effectiveness of traditional and cloud based hosting. We have examined three different websites in diversified environments. During performance and resource consumption tests we were able to demystify some myths that were considered as a common knowledge, but also managed to generally examine the applications running on systems with multicore processors. Analysis of the results allowed us to draw several noteworthy conclusions.

The research confirmed the fundamental belief stating that the choice of nowadays web hosting is not a trivial task. However, it denied our expectation that both hosting types would deliver comparable results, at least for the tested web applications. Some believes that cloud based hosting is far superior in many ways; we can confirm that in terms of resilience and availability it has no match; however, an intricate model of charging and high ongoing costs make it still economically unjustified. The nuisance in choice between two ostensibly alike hosting types, generally lies in finding a common ground of comparison. Even if just a cost and technical parameters are set as a common denominator, adequate comparison constitutes a bit of challenge. Monthly (and annual) total cost of hosting was primary (and decisive) factor in our final assessment of traditional and cloud based hosting. Traditional hosting proved to be a better choice for two of three types of the surveyed sites (*Standard* and *Business*). The cheapest proposed solution (a single, and yet not very powerful VPS) on a dedicated server was able to handle a larger number of active users than the most expensive instance of the cloud. The only exception was a *brochure* for which both traditional and cloud based hosting were able to handle the same maximum number of users at the same relatively inexpensive, but the most affordable hosting for it, turned to be cloud (it was just marginally less expensive). Situation would change diametrically if database would have been used, which in cloud based hosting involves additional costs. That was the main reason why for two other types of tested websites was cheaper to host them on a dedicated server regardless of a time-frame. In terms of technical parameters, it is worth to consider two things: even for websites with statistically little traffic the memory size should be greater than 2 GB (not enough memory is usually a bottleneck), whereas number of CPUs (and their speed) is vital for the websites with high traffic. The achieved results have raised new ideas for research. It would be useful to prepare more diversified sites. Architecturally and in terms even more mixed traffic. It would be useful to analyze the effect of streaming services and those who need large resources only in specific and custom time intervals. It would also good to break the limitation of a Web server. The point here is not being limited by a number of active requests, and to be able ALWAYS fulfil more requests with a stronger configuration. In the case of services based on PHP (like our *Standard*/*Business* sites), it would be beneficial for the sake of performance to test *nginx* server.

References

1. Bellavista, P.: A practical approach to easily monitoring and managing IaaS environments. In: 2013 IEEE Symposium on Computers and Communications (ISCC)
2. Bhardwaj, S., Jain, L., Jain, S.: Cloud computing: a study of infrastructure as a service (IAAS). Int. J. Eng. Inf. Technol. **2**(1), 60–63 (2011)
3. Grossman, R.L.: The case for cloud computing. IT Prof. **11**(2), 23–27 (2009)
4. Molnar, D., Schechter, S.E.: Self hosting vs. Cloud hosting: accounting for the security impact of hosting in the cloud. In: WEIS, June 2010
5. Rimal, B.P., Choi, E., Lumb, I.: A taxonomy and survey of cloud computing systems. In: INC, IMS and IDC, pp. 44-51 (2009)
6. Roustemaa, J.: Cloud Server vs. VPS vs. Dedicated Server, August 2015. https://www.upcloud.com/blog/cloud-server-vs-vps-vs-dedicated-server/
7. Strebel, J., Stage, A.: An economic decision model for business software application deployment on hybrid Cloud environments. In: Multikonferenz Wirtschaftsinformatik 2010, p. 47 (2010)
8. Zhou, M., Zhang, R., Zeng, D., Qian, W.: Services in the cloud computing era: a survey. In: 2016 4th International Universal Communication Symposium (IUCS), pp. 40–46. IEEE, October 2010
9. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. **1**(1), 7–18 (2015)
10. Internet live stats. http://www.internetlivestats.com/one-second/
11. Web Server Survey, January 2017. https://news.netcraft.com/archives/category/web-server-survey/

12. Usage of content management systems for websites. https://w3techs.com/technologies/overview/content_management/all
13. Nine Most Popular Free Content Management Systems. https://colorlib.com/wp/most-popular-content-management-systems/
14. Cloud hosting vs traditional hosting. http://www.opusinteractive.com/cloud-hosting-vs-traditional-hosting/
15. Comparing cloud vs on-premise. Six Hidden Costs People Always Forget About. https://betanews.com/2013/11/04/comparing-cloud-vs-on-premise-six-hidden-costs-people-always-forget-about/
16. https://gigaom.com/2012/02/11/which-is-less-expensive-amazon-or-self-hosted/
17. https://aws.amazon.com/blogs/aws/be-careful-when-comparing-aws-costs/
18. Gartner, I.T.: Glossary > Infrastructure as a Service (IaaS). http://www.gartner.com/it-glossary/infrastructure-as-a-service-iaas/
19. Cloud computing - statistics on the use by Enterprises (2014). http://ec.europa.eu/eurostat/statistics-explained/index.php/Cloud_computing_-_statistics_on_the_use_by_enterprises
20. RightScale, State of the Cloud Report. https://www.rightscale.com/lp/state-of-the-cloud?campaign=7701700000015ln8
21. Web Hosting, The most comprehensive Web Hosting statistics in the world. http://webhosting.info/web-hosting-statistics

High Quality Stabilization of an Inverted Pendulum Using the Controller Based on Trigonometric Function

Michał Lower[✉]

Wroclaw University of Science and Technology,
ul. Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
michal.lower@pwr.edu.pl

Abstract. Inverted pendulum is an example of a model that is used to simulate many phenomena eg. quiet standing. Because it is a nonlinear object and it is difficult to stabilize, new ways of controlling and stabilizing the inverted pendulum are constantly being sought. Finding these methods is very important in the reliable operation of the whole system of which the inverted pendulum is a part. It should be emphasized that the inverted pendulum generally has its critical point beyond which it falls over, which makes the further work of the system impossible. This paper shows a new approach to stabilization it using trigonometric functions that are oscillating. The presented solution is characterized by high stabilization efficiency simultaneously for small and large swinging of the pendulum. The concept of stabilizing the pendulum is based on the similarity of coping with this problem by human individual. The individual without knowing the mathematical model of the object is able to stabilize the inverted pendulum by oscillating movements, changing the amplitude and the oscillation period. Such an action can be described by a trigonometric function.

Keywords: Inverted pendulum · Stabilization and control · Nonlinear systems

1 Introduction

Many examples of processes that use the inverted pendulum to model its components can be found in the literature. Such solutions as a Segway, which is a balancing robot, are widely known [1,3,11]. An inverted pendulum model can be used to study in quiet standing [8]. Inverted pendulum may also be an introduction to more complex calculations such as stabilization of helicopter in hover [6,7] or other multirotors such as a quadrotor [9,14].

The basic regulator used to stabilize the inverted pendulum is PID [10,12], but inverted pendulum is nonlinear and difficult to stabilize, hence new methods of control and stabilization are constantly sought. Finding these methods is very important in the reliable operation of the whole system of which the inverted

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_24

pendulum is a part. It should be emphasized that the inverted pendulum usually has its critical point beyond which it falls over, which prevents the further work of the system. Scientists are looking for new and better solutions. Fuzzy logic [4,8,13], artificial neural networks [5,15], an inverted model [2] are used to control. The paper presents a new approach to stabilization using trigonometric functions that are oscillating. Studies show that the solution is characterized by high stability at the same time for small and large pendulum swings. The concept of stabilizing the pendulum is based on the similarity of coping with this problem by human individual. The individual without knowing the model of a mathematical object is able to master the stabilization of the inverted pendulum by means of oscillating movements, changing the amplitude and oscillation period. Such an action can be described by a trigonometric function.

2 The Simulation Model of the Pendulum

One of the processes that use inverted pendulum for process modeling is quiet standing [8]. For the modeling of the inverted pendulum in this process, a system is used where the base is still on the ground and the pendulum has the ability to rotate in two planes separately stabilized by the same algorithm. The movement of the pendulum in a single plane can be described in accordance with Fig. 1. The pendulum has the ability to deviate from the vertical by an angle φ. For the purposes of the test, it can be assumed that the angle φ takes values in a range from +90° to −90°. Stabilization of the position of the pendulum is obtained by forcing the straightening torque M, where M equals the product of the forcing force F_x and the radius (the pendulum arm) R.

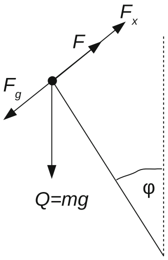


Fig. 1. The model of inverted pendulum.

Model parameters:

m – mass of inverted pendulum,
R – radius (the pendulum arm), distance from the center of mass to the point - the center of rotation,
φ – angle of the pendulum deviation from the vertical,
ω – angular velocity of the pendulum,

ε – angular acceleration,
g – gravitational acceleration,
a – linear acceleration of the center of mass of the pendulum, acceleration vector perpendicular to the radius R,
Q – force of gravity, therefore $Q = mg$,
F_x – force for stabilizing the pendulum.

Force of gravity Q is distributed to the pressure on the basis of the pendulum and the force F_g, therefore (1).

$$F_g = Q\sin(\varphi) = mg\sin(\varphi) \tag{1}$$

while the force for stabilizing the pendulum is (2):

$$F = -am = -\varepsilon Rm \tag{2}$$

The resultant force F influencing the movement of the pendulum is described by the formula (3). As a result, we obtain the equation of the pendulum (4) and (5).

$$F = F_x - F_g \tag{3}$$

$$-am = F_x - mg\sin(\varphi) \tag{4}$$

$$\varepsilon Rm = mg\sin(\varphi) - F_x \tag{5}$$

The simulation model of an inverted pendulum was made on the basis of the formula (5) as shown in Fig. 2, in MatLab Simulink.

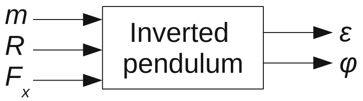


Fig. 2. The simulation block of the inverted pendulum.

The input vector of model consists of three variables:

m – mass of the pendulum, variable in the formula denoted as m, the first input of block,
R – length of pendulum arm, the second input of block,
F_x – force for stabilizing the pendulum, variable in the formula denoted as F_x, the third input of block.

The output vector of the model consists of two variables:

ε - angular acceleration, the first output of block,
φ - angle of pendulum deviation from the vertical, the second output of block.

3 Inverted Pendulum Controller and Test Results

Human individual using oscillating movements, changing the amplitude and oscillation period is able to effectively stabilize the pendulum. By observing this way of dealing with the problem, controllers using the trigonometric function can be developed. The paper presents two trigonometric controllers. The first one is a simplified controller, while the second one is expanded and it is called a complete controller.

The simplified controller is conforming to the formula (6). The controller has one parameter to set k_s coefficient. For the tests the value of k_s coefficient was assumed to be 1000.

$$F_x = k_s \left(2g\sin\left(\frac{\varphi}{2}\right) + 9.81\omega\right) \tag{6}$$

The complete controller is conforming to the formula (7). The controller has one parameter for the setting of k_c coefficient, which was set to 300 in the tests. The study found that the sine function and the angular velocity should be raised to the square, but the negative values of these elements should be maintained. Hence, absolute values were introduced in the formulas. Similarly, it was found that the value of general amplification of the formula should be inversely proportional to the absolute value of the deviation angle.

$$F_x = \frac{k_c}{|\varphi|}\left(2g\sin\left(\frac{\varphi}{2}\right)\left|\sin\left(\frac{\varphi}{2}\right)\right| + \frac{R}{2}\omega\,|\omega|\right) \tag{7}$$

In the studied models, the control system was designed to bring the inverted pendulum to a state of equilibrium, i.e., φ equals to zero. In the inverted pendulum model, the initial value of the angle φ of the deviation from the vertical is introduced. The initial value of the deviation is set to 1 rad, which is approximately 57.3°. In the model of the inverted pendulum the values $m = 80\,\mathrm{kg}$ and $R = 1.2\,\mathrm{m}$ were assumed.

The tests were carried out in a direct connection of the controller and the inverted pendulum, according to the diagram in Fig. 3 and in the system with a distorting inertial block, as shown in the diagram in Fig. 4. Transfer function $Fc1$ of inertial block of executive system is compatible with the model (8)

$$Fc1 = \frac{1}{0.0025s^2 + 0.1s} \tag{8}$$

3.1 The Symulation Tests of the Simplified Controller

The simulation results of the simplified control system in the direct connection of the controller and the inverted pendulum are shown in Fig. 5. The setpoint has been reached in less than 15 s without visibly exceeding the set point. The simulation results of the simplified control system in combination with the inertial block are shown in Fig. 6. After this change the stabilization was not achieved.

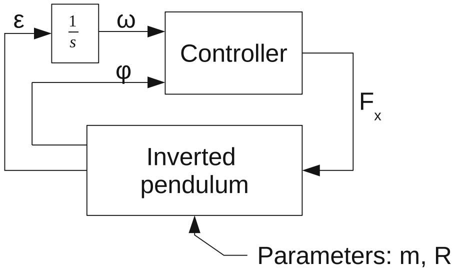


Fig. 3. The scheme of the simulation system without inertial function.

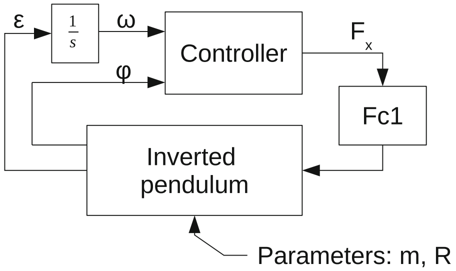


Fig. 4. The scheme of the simulation system with inertial function.

Keeping in mind that the inverted pendulum is a non-stationary and nonlinear object, it should be considered that the direction of research is appropriate. Simple oscillatory system very well adjusts the pendulum to the set point, however, this proposal is not resistant to interference, which may occur in the real object.

3.2 The Symulation Tests of the Complete Controller

The simulation results of the complete control model in the direct connection of the controller and the inverted pendulum are shown in Fig. 7, while the system connected to the inertial block $Fc1$ is shown in Fig. 8. In both cases, the system has been adjusted to a setpoint value in less than 5 s without exceeding the setpoint. The influence of the inertial part on the adjustment process is unnoticeable.

3.3 The Symulation Tests of the Complete Controller for the Changed Object Parameters – The Change of Mass

In another test of the complete control model of inverted pendulum, attention was paid to its sensitivity to the change in mass of the pendulum. During the test the same initial conditions and all controller settings were set. The inertial function $Fc1$ (Fig. 4) was also used. The mass of the pendulum was reduced by half, instead of $m = 80\,\text{kg}$ it was set to $m = 40\,\text{kg}$.

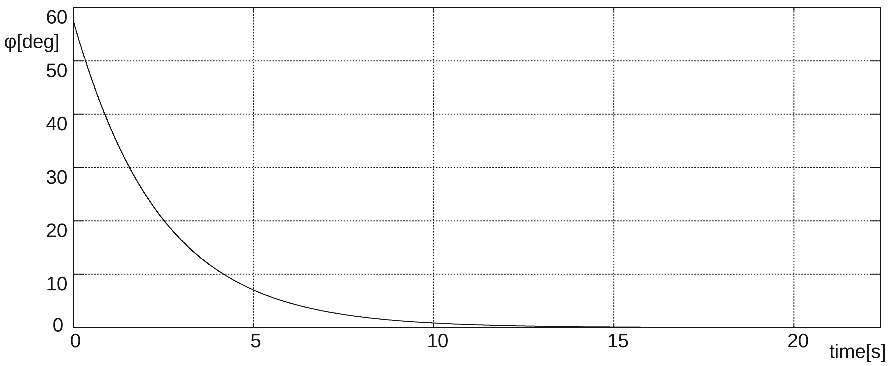


Fig. 5. The simulation results of system without inertial function – the simplified controller

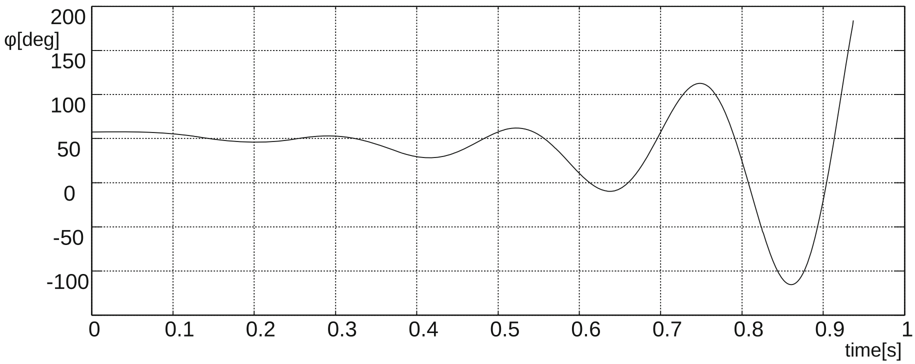


Fig. 6. The simulation results of system with inertial function – the simplified controller

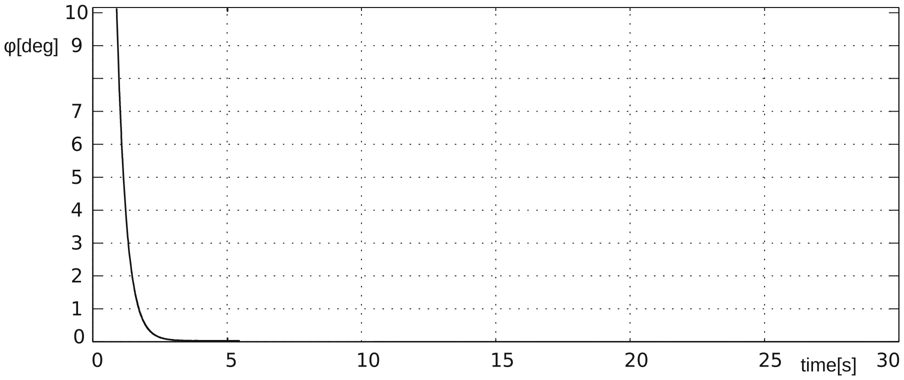


Fig. 7. The simulation results of system without inertial function – the complete controller

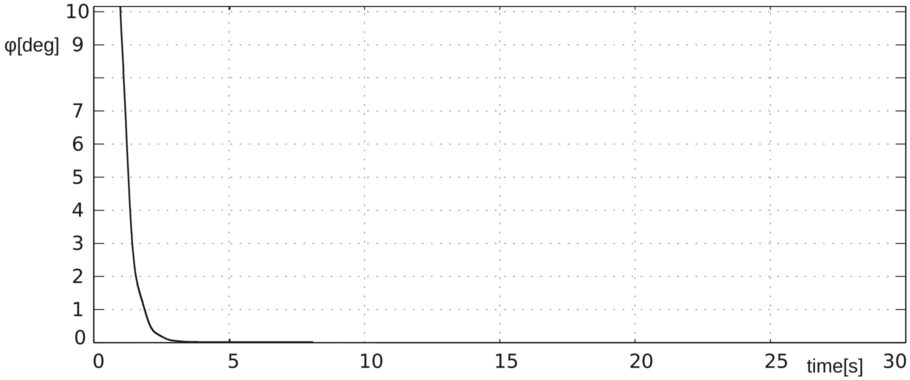


Fig. 8. The simulation results of system with inertial function – the complete controller

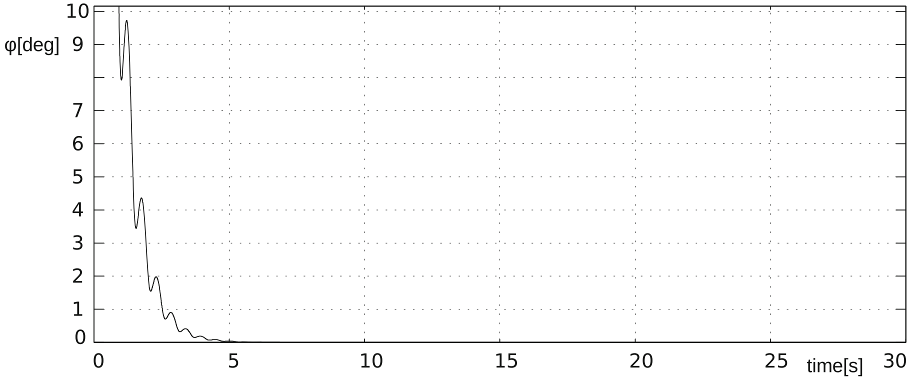


Fig. 9. The simulation results of system with inertial function – the complete controller and $m = 40\,\mathrm{kg}$

As shown in the diagram in Fig. 9, the system has been adjusted to a setpoint value of less than 5 s, without visibly exceeding the setpoint. The effect of mass change on the control process is shown in the diagram in the form of minor disturbances, but it did not significantly affect the stability of the system.

3.4 The Symulation Tests of the Complete Controller in Response to Small Disturbances

The complete control model was tested for small disturbances. Sometimes, classical regulators tuned to large deviations, have a poor quality control with small disturbances. During the test, the same initial conditions and all controller settings were set as in the previous tests. The inertial function $Fc1$ (Fig. 4) was also used. Tests were performed for two values of the pendulum mass $m = 80\,\mathrm{kg}$ and $m = 40\,\mathrm{kg}$.

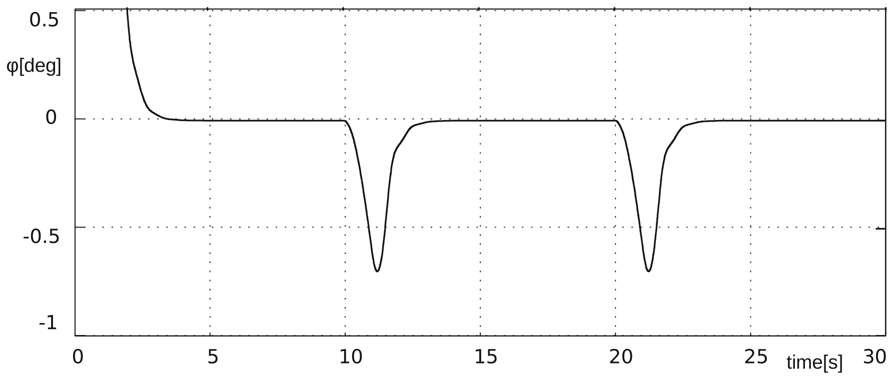


Fig. 10. The simulation results of system with inertial function and small disturbances – the complete controller and $m = 80\,\mathrm{kg}$

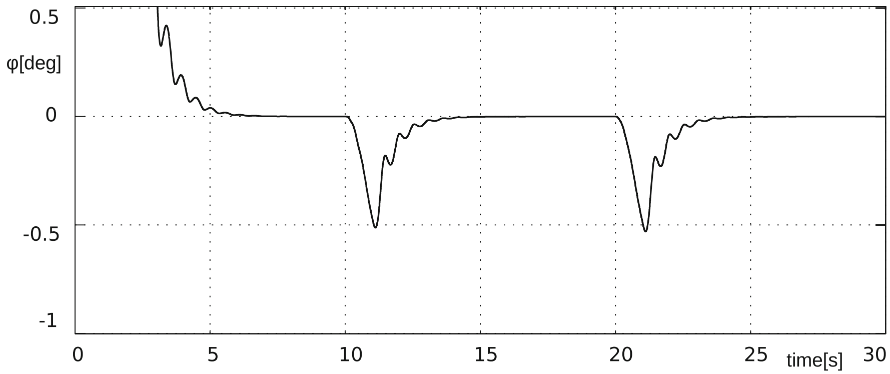


Fig. 11. The simulation results of system with inertial function and small disturbances – the complete controller and $m = 40\,\mathrm{kg}$

The disturbance was introduced into the F_x control value. The control value was added to the external 10 N interfering force, occurring for 1 s at intervals of 10 s.

The results of the tests are shown in Fig. 10 for a pendulum of mass $m = 80\,\mathrm{kg}$ while for a pendulum of mass $m = 40\,\mathrm{kg}$ in Fig. 11. In both cases, the system was adjusted to the setpoint in about 3 s without exceeding the setpoint and without oscillation. The complete control model effectively responds to small disturbances.

4 Conclusions

The conducted simulation studies have demonstrated the high quality of the inverted pendulum stabilization for large deviations and small disturbances.

Thus, the approach to stabilization using trigonometric functions is the appropriate direction. In classic solutions such as PID, the controller is tuned to a particular work point, beyond which the new controller parameters are required to be set. In nonlinear systems, such as inverted pendulum, the change of work point, e.g. the change of mass (especially with such a high value from 80 kg to 40 kg) or work range, can destabilize the entire system. Sometimes the operating point of the pendulum may change due to minor damage, malfunction, or operational reasons. Thus, the proposed solution can significantly increase the reliability of the entire system. The presented model of inverted pendulum was developed on the model of individual in quiet standing. Bearing this aspect in mind, further studies of the human model in quiet standing can be carried out using the controller proposed in the paper.

The scope of further research can be focused on the development of the model of the inverted pendulum and systems similar to this model. An inverted pendulum control model for the trolley can be developed, taking into account the moment of inertia of the pendulum. In further research, the derived control principles in the helicopter autopilot can be checked out.

It should be emphasized that the proposed model of controller has only one parameter to set, so tuning such a controller is simple and quick to implement by empirical method.

References

1. Blackwell, T., Casner, D., Wiley, S.: Remotely controlled self-balancing robot including kinematic image stabilization, 20 May 2014. US Patent 8,731,720. https://www.google.com/patents/US8731720
2. Boussaada, I., Morarescu, I., Niculescu, S.: Inverted pendulum stabilization: characterization of codimension-three triple zero bifurcation via multiple delayed proportional gains. Syst. Control Lett. **82**, 1–9 (2015)
3. Brning, M., Schnewolf, W., Krger, J.: Stabilisation and manoeuvre of electrically powered pedestrian controlled uniaxial vehicles for goods transport. In: 2014 UKACC International Conference on Control (CONTROL), pp. 31–38, July 2014
4. Dang, Q.V., Allouche, B., Vermeiren, L., Dequidt, A., Dambrine, M.: Design and implementation of a robust fuzzy controller for a rotary inverted pendulum using the takagi-sugeno descriptor representation. In: 2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA), pp. 1–6 (2014)
5. Krafes, S., Chalh, Z., Saka, A.: Review: linear, nonlinear and intelligent controllers for the inverted pendulum problem. In: 2016 International Conference on Electrical and Information Technologies (ICEIT), pp. 136–141 (2016)
6. Król, D., Lower, M., Szlachetko, B.: Selection and setting of an intelligent fuzzy regulator based on nonlinear model simulations of a helicopter in hover. New Gener. Comput. **27**(3), 215–237 (2009)
7. Lower, M., Szlachetko, B., Krol, D.: Fuzzy flight control system for helicopter intelligence in hover. In: 5th International Conference on Intelligent Systems Design and Applications (ISDA 2005), pp. 370–374 (2005)
8. Lower, M.: Simulation model of human individual in quiet standing based on an inverted pendulum with fuzzy controller. In: 2008 International Conference on Machine Learning and Cybernetics, vol. 6, pp. 3418–3422, July 2008

9. Lower, M., Tarnawski, W.: Quadrotor Navigation Using the PID and Neural Network Controller, pp. 265–274. Springer, Cham (2015)
10. Mandić, P.D., Lazarević, M.P., Šekara, T.B.: Stabilization of Inverted Pendulum by Fractional Order PD Controller with Experimental Validation: D-decomposition Approach, pp. 29–37. Springer, Cham (2017)
11. Nagarajan, U., Yamane, K.: Universal balancing controller for robust lateral stabilization of bipedal robots in dynamic, unstable environments. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 6698–6705, May 2014
12. Prasad, L.B., Tyagi, B., Gupta, H.O.: Optimal control of nonlinear inverted pendulum system using pid controller and lqr: performance analysis without and with disturbance input. Int. J. Autom. Comput. **11**(6), 661–670 (2014)
13. Raj, S.: Reinforcement learning based controller for stabilization of double inverted pendulum. In: 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), pp. 1–5, July 2016
14. Szlachetko, B., Lower, M.: Stabilisation and Steering of Quadrocopters Using Fuzzy Logic Regulators, pp. 691–698. Springer, Heidelberg (2012)
15. Wang, Y., Shen, H., Duan, D.: On stabilization of quantized sampled-data neural-network-based control systems. IEEE Trans. Cybern. **4**(99), 1–12 (2016)

The Application of RFID Technology in Supporting the Process of Reliable Identification of Objects in Video Surveillance Systems

Piotr Lubkowski[1](✉), Dariusz Laskowski[1], and Marcin Polkowski[2]

[1] Military University of Technology, Gen. S. Kaliskiego 2 Street,
00-908 Warsaw, Poland
{piotr.lubkowski,dariusz.laskowski}@wat.edu.pl
[2] Transbit Sp. z o. o., Łukasza Drewny 80 Street, 02-968 Warsaw, Poland
marcin.polkowski@transbit.com.pl

Abstract. A process of ensuring the security of citizens requires an access to the information from sensors located in different points of monitoring and data acquisition systems [1]. Automating the process of detection and identification of people that uses marking objects and their identification on the basis of defined database is crucial for enhancing the level of security of protected resources. The paper proposes a concept of application that combine the advantages of RFID (*Radio-Frequency Identification*) and technique of recognition of persons on the basis of facial features. The proposed solution makes it possible to increase the efficiency and reliability of the monitoring system in the field of rapid and unambiguous identification of system users [2].

Keywords: Video monitoring · Reliable identification · RFID technology

1 Introduction

Video monitoring systems have been used for many years to increase security level of citizens and protected facilities. We encounter video surveillance in both public and national utility facilities, in commonly available places and in the areas with limited access. Monitoring systems are a combination of video recording (sensors), transmitting, storing and reproducing devices in one integral unit. They enable observation of people or objects in real time as well as event recording for later analysis. Different kinds of monitoring services in the technical environment have been described in [2, 3].

The detection and identification of people/objects is one of the primary advantages offered in the modern surveillance systems. Automatic detection and recognition of people based on facial features is an active area of research covering different fields of science [4]. However, in order to obtain such functionality several factors that affect the efficiency of the process of analysis, detection and identification of facial features require consideration. In the reliable systems, the following factors should be taken into account: lack of proper object lighting (insufficient sensitivity of the transducer) or no infrared operation mode, too long distance between the object and the camera

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_25

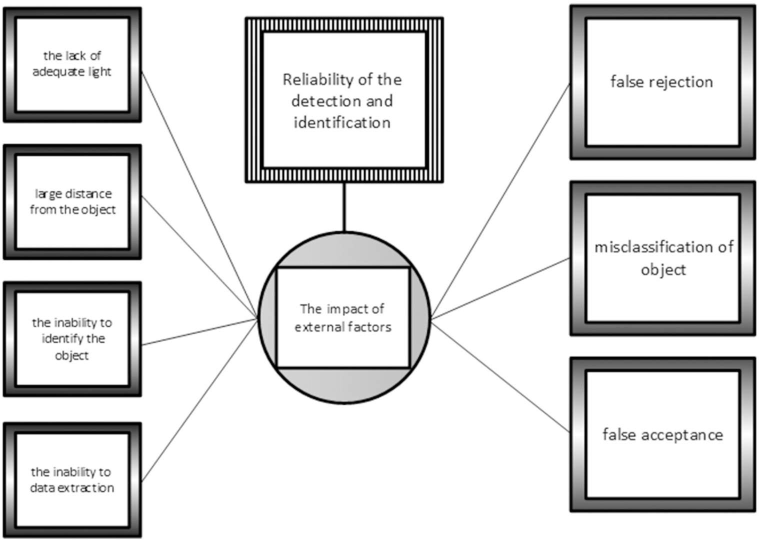

the lack of adequate light
large distance from the object
the inability to identify the object
the inability to data extraction
Reliability of the detection and identification
The impact of external factors
false rejection
misclassification of object
false acceptance

Fig. 1. The Bow Tie analysis of object identification problems in terms of the influence of external factors

(no proper selection of the focal length), no possibility of object specification (insufficient sensor resolution) or no extraction possibility (low resolution, sensitivity). The impact of environmental and sensor technical factors on reliability of the detection and quality of identification processes has been discussed in [5]. Each of these degrading factors effect on the reliability and speed of the identification process, which are a critical indicator of the system's readiness for common use. They can lead to the following problems associated with the reliable identification process (Fig. 1):

- false rejection - an object that has its model in the database is unrecognized and rejected due to the fact that it does not have its counterpart,
- misclassification - an object that has its model in the database is not properly assigned to other model in the database,
- false acceptance - an object that does not have its model in the database is assigned to a model that already exists in the database.

In order to minimize the impact of mentioned degrading factors, in particular the time of identification, the monitoring systems increasingly use solutions that support the automation of the verification process. An example of such a solution may be the RFID technology, in which the identification is carried out through the exchange of data between the tag and the reading device using radio waves.

The RFID technology is currently one of the most widely used techniques for automatic identification. This is due to the following facts:

- a global standardization, which enables the use of various patterns on a larger scale,
- improving the process of entering data, relevant to the verification of a large number of objects,

- the ability to re-write and modify data in the tag,
- an automated readout which does not require human intervention,
- a data security,
- the IDs resistance to damage and their long service life,
- the possibility of integration with other systems of automatic identification,
- shortening the time of objects verification.

A number of examples of complex monitoring systems using RFID technology can be found in the literature [6, 7]. In the majority of cases passive tags are applicable, which are used to identify objects in the processes of logistics, transportation, manufacturing, monitoring of supplies and support for navigation. An access control system combining RFID technology and face recognition based on neural network is presented in [8]. The system recognizes the face of the person holding the RFID card and denies access if they do not match. Wide discussion of different attendance management systems is given in [9]. Some advantages and disadvantages of Bluetooth, Iris, fingerprint, face recognition and RFID solutions are explained as well. The [10] shows an embedded face recognition authentication system, which consists of the RFID card used to store the face *eigen* information and the face recognition unit. The main benefit of this hardware solution is shortness of time for identification. An increasing a reliability and speed of object identification was also the basic assumption taken into account in the process of developing the *RFID_FACE_REC* application.

The paper presents the innovative concept, functional description and implementation of application utilizing RFID technology in supporting the process of people identification in the video surveillance system. The identification is carried out in two stages, and combines the advantages of RFID technology and automatic detection and faces recognition in a digital video image (Fig. 2). Presented application was practically implemented in a video surveillance network of Telecommunications Institute laboratory.

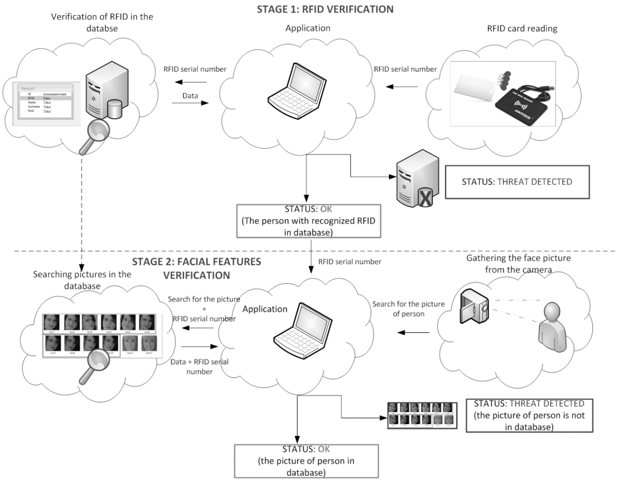

STAGE 1: RFID VERIFICATION
Verification of RFID in the databse
Application
RFID card reading
RFID serial number
Data
RFID serial number
STATUS: THREAT DETECTED
STATUS: OK (The person with recognized RFID in database)
STAGE 2: FACIAL FEATURES VERIFICATION
Searching pictures in the database
RFID serial number
Gathering the face picture from the camera
Search for the picture + RFID serial number
Application
Search for the picture of person
Data + RFID serial number
STATUS: THREAT DETECTED (the picture of person is not in database)
STATUS: OK (the picture of person in database)

Fig. 2. Diagram of a two-step identification of persons with the use of RFID and facial recognition

2 The Concept and Implementation of RFID_FACE_REC Application

2.1 Functionality of the Application

As it was already mentioned, the concept of *RFID_FACE_REC* application is based on implementation of two essential functions comprising identification of the object based on the assigned RFID value followed by the identification based on the facial features recognition. Application of these functionalities in conjunction with the fulfillment of specific technical and operational requirements determines its effectiveness and reliability. The essential functional requirements defined with reference to the presented application are presented in Fig. 3 while the operational and technical requirements are shown in Fig. 4.

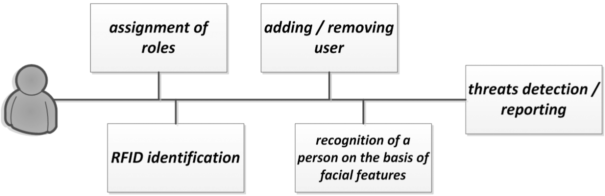

assignment of roles
adding / removing user
threats detection / reporting
RFID identification
recognition of a person on the basis of facial features

Fig. 3. The functional requirements for *RFID_FACE_REC* application

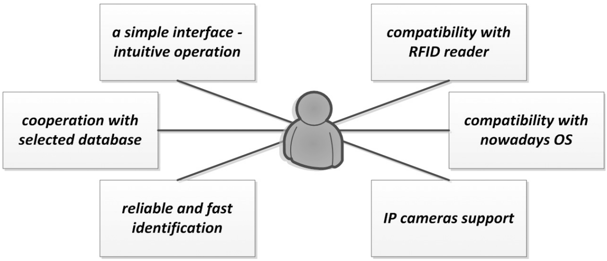

a simple interface - intuitive operation
compatibility with RFID reader
cooperation with selected database
compatibility with nowadays OS
reliable and fast identification
IP cameras support

Fig. 4. Technical and operational requirements for *RFID_FACE_REC* application

Fast and unambiguous identification of persons, which allows immediate reporting of threats in the monitored object is particularly important. The cooperation of application with the family of IP cameras as well as a wide range of RFID tags makes it possible to use it in a complex video surveillance system. Furthermore, through the use of graphical interface, it is possible to signal detected threats using “pop-up” windows that are activated automatically after completion of identification. The application provides interaction with the Microsoft Office Access database, which offers access to the information about user permissions.

2.2 Implementation of the Application

The RFID*_FACE_REC* application, which is the basis of the measurement system, have been realized on the basis of Microsoft Visual Studio Community 2015 development environment and consists of two software modules written using C# language. The first module is responsible for the verification of the object using the RFID technology. For verification purposes KT-STD-2 passive transponders have been used, operating at a frequency of 125 kHz, which in its data memory has saved the unique number assigned by the manufacturer. The UNIQUE readers of JABLOTRON company (modelJA-190T) cooperate with RFID transponders. They also operate at a frequency of 125 kHz. The task of the reader is supplying power for the transponder and reading the *Transponder Identification Number* (TID). The reader cooperates with a personal computer through the USB interface. The data read from the reader are then compared with the information stored in the database. In this way, the first step of identification is carried out.

A graphical interface built on the activities related to the servicing start-up screen of application (*User.cs*), menu for user creation (*AddUser.cs*), menu for graphical visualization of data (*Form2.cs*) and the main menu of the application (*RfidForm.cs*), which is responsible for the first stage of identification, have been developed for the communication with the user. Communication with the database uses the OLEDB programming interface developed by Microsoft. It guarantees universal access to data sources, regardless of the form and ways of storing them.

The process of retrieving data for a specific RFID number is implemented using *RfidForm.cs* class. The result of SQL (*Structured Query Language*) query is transferred to the *personData* object and is then used by the module responsible for the identification of a person based on the analysis of facial features. This module uses the modified for the project purposes a "*Multiple face detection and recognition in real time*" (MFRRT) application described in [11]. The application mentioned in the process of detection and face recognition uses a method called "*own faces*" that utilizes *Principal Component Analysis* (PCA). In the process of recognition an *EmguCV* library is used [12], which offers a number of important functions related to the detection and identification, including character recognition, face and motion of objects detection. The modifications consist in adding new functionalities. The first enables adding a new user's face into database and linking it with the declared number of RFID (*Class Form2.cs*). User's picture (in the format of *bmp*) and the associated number of RFID are stored in the root directory of the "*TrainedLabels*" project with a unique identifier. The second function is responsible for face recognition of an identified person and check whether it is associated with a previously verified RFID card. This operation is supported with the *MainForm.cs* class that is responsible for generation of the graphics window informing the system administrator about result of recognition and verification.

The application cooperates with family of IP cameras, whose main characteristics are shown in Table 1. After running the main window of application is called in a position of waiting for scanning RFID cards.

Table 1. Features of the used IP cameras

Model of IP cameras	The image resolution
Axis M1031-W	640 × 480
Axis Q1755	1280 × 720
Axis 207 MW	640 × 480
Axis 215 PTZ	702 × 576

3 Testing and Analyzing the Results

For the experiment, the existing infrastructure of the laboratory monitoring network has been used (Fig. 5) [1, 2]. Reasoning about the correctness of network components specification as regards data transmission reflecting information from the monitoring system is based on a statistical estimation of reliability of the software and hardware platform forming the service chain. The presented infrastructure is also a result of our experience gained when designing a laboratory network environment, as well as the conclusions of the analysis of the literature [3, 5]. Products of renowned suppliers of hardware and software for both the systems and applications are the components of the test platform. Therefore, concluding that the specified measuring system is a correct and highly reliable testbed seems to be reasonable.

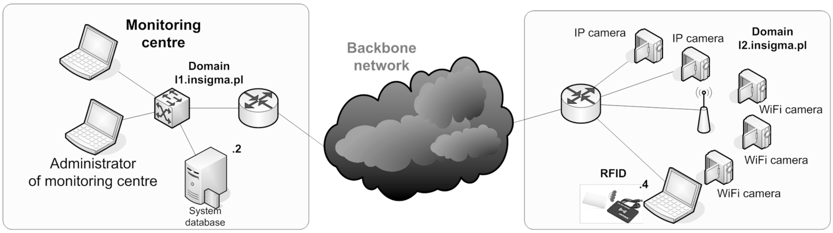
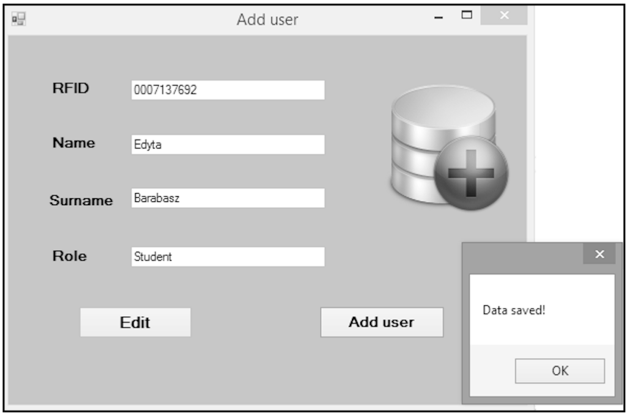


Fig. 5. Diagram of the test bed environment

The presented system (Fig. 5) was supplemented with elements used in the verification process by means of RFID technology, which are already mentioned passive transponders KT-STD-2 and the RFID reader (model type JA-190T). The position of administrator (located in the domain l1.insigma.pl) was represented by PC laptop equipped with Intel Core i3-3217CPU 1.180 GHz and working over the Windows operating system. For communication with the camera Real-Time Streaming Protocol (RTSP) is used.

Commissioning tests were related to the verification of the possibility of implementation and starting the application and determination of the correctness of its basic functionalities. The application offers the user access to options supporting the process of verification and identification as defined in the set of presented functional, operational and technical requirements. The administrator gets an opportunity to create the user based on the definition of an RFID tag, and create a database of face images (Fig. 6).



RFID 0007137692

Name Edyta

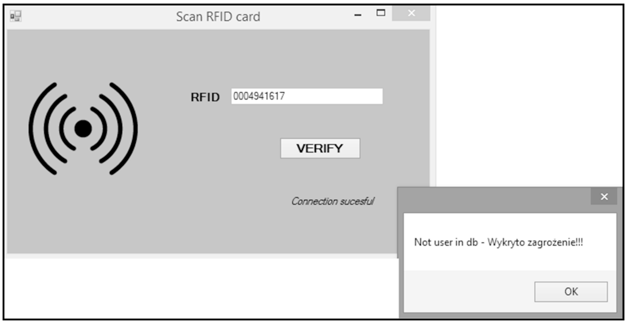
Surname Barabasz

Role Student

Data saved!

Fig. 6. User creation menu

Another useful feature is the verification and identification mode of application work where the ability to detect threats at the stage of verification of RFID (Fig. 7) or at the stage of identifying facial features (Fig. 8) is offered.

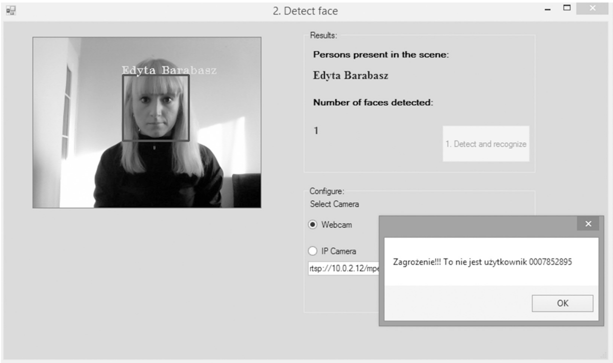


RFID 0004941617

Connection sucesful

Not user in db - Wykryto zagrożenie!!!

Fig. 7. The threat detected in the first stage of recognition



Results:

Persons present in the scene:

Edyta Barabasz

Number of faces detected:

1

Configure:

Select Camera

● Webcam

○ IP Camera

rtsp://10.0.2.12/mpe

Zagrożenie!!! To nie jest użytkownik 0007852895

Fig. 8. Detection of threats - a person from the database with a different number of RFID

The studies related to the determination the correctness of the user verification process using the RFID reader and determination the effect of the number of images in the database and camera resolution on the correctness of identification were carried out within the functional tests. In order to evaluate the results the criterion of proper objects identification was determined. This criterion has been established in the form of recognition probability:

1. sufficient at least equal to 50%;
2. satisfying at least equal to 60%;
3. correct at least equal to 70%;
4. suitable at least equal to 90%.

The results of the experiments presented in the figure (Fig. 9) show the correctness of identification in a function of number of images and camera resolution. During the

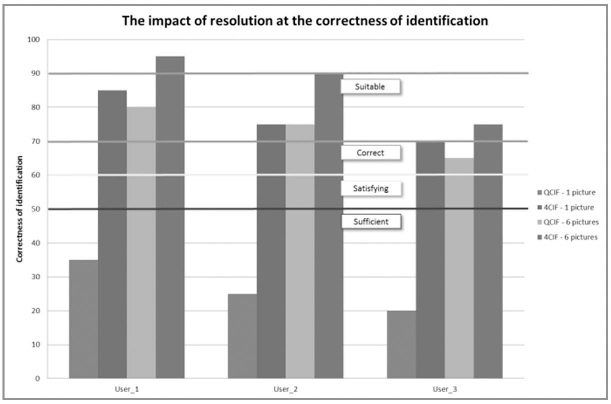


Fig. 9. The correctness of identification as a function of the number of images in the database and camera resolution

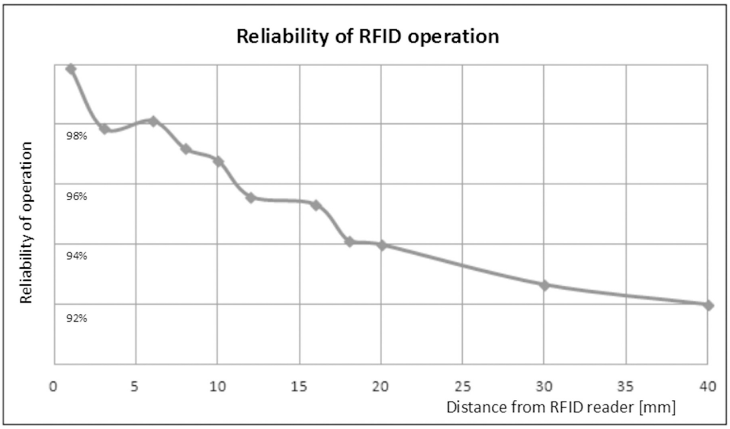


Fig. 10. The reliability of RFID tag operation as a function of distance from RFID reader

tests the camera worked in a low resolution mode (QCIF – 176 × 144), and then was switched to the of high resolution mode (4CIF – 704 × 576). The database contained respectively 1 or 6 images of recognized person. It should be noted that the correct identification is possible even at low resolution of camera. It is also worth to emphasize that for the first stage of the application work, i.e. verification of RFID, an almost 100% correctness of recognition was observed (Fig. 10).

The use of this variant of work does not provide however the 100% guarantee of correct operation of the system.

4 Summary

The obtained results confirm the accomplishment of assumptions set for the *RFID_-FACE_REC* applications in order to improve the reliability and speed of people identification in video surveillance systems.

The results of the tests regarded to the application RFID module indicates the independence of verification from the test environment and used sensors. Each of the tests performed for UNIQUE standard has confirmed the correctness of object identification that was based on the prepared database. This ensures a high level of security of the protected object. Proper verification of the object in the first phase enables the realization of the second stage, i.e. the identification of the person based on facial features. In case of positive result of identification the transmission of information about absence of any threat is realized. Otherwise, the alarm about the threat for the controlled area or object is generated to the security.

In summary it is worth noting, that the use of a hybrid combination of two mentioned techniques of automatic identification improves the reliability of the system, reduces the time required for identification and enhances safety. It should be also mentioned that the presented application, at the current stage of development, does not aspire to be a professional application, but its purpose is to estimate the possibility of using it in specific environmental and technical conditions, in order to obtain conclusions for further development work. The obtained results are determinant for further studies in the area of surveillance systems.

References

1. Lubkowski, P., Laskowski, D., Maślanka, K.: On supporting a reliable performance of monitoring services with a guaranteed quality level in a heterogeneous environment. In: Theory and Engineering of Complex Systems and Dependability, vol. 365, pp. 275–284. Springer, Heidelberg (2015)
2. Lubkowski, P., Laskowski, D.: Test of the multimedia services implementation in information and communication networks. In: Advances in Intelligent Systems and Computing, vol. 286, pp. 325–332. Springer, Heidelberg (2014)
3. Siergiejczyk, M., Paś, J., Rosiński, A.: Application of closed circuit television for highway telematics. In: The Monograph Telematics in the Transport Environment. CCIS, vol. 329, pp. 159–165. Springer, Heidelberg (2012)

4. Siergiejczyk, M., Krzykowska, K., Rosiński, A.: Reliability assessment of cooperation and replacement of surveillance systems in air traffic. In: Advances in Intelligent Systems and Computing, vol. 286, pp. 403–411. Springer, Heidelberg (2014)
5. Lubkowski, P., Laskowski, D., Pawlak, E.: Provision of the reliable video surveillance services in heterogeneous networks. In: Safety and Reliability: Methodology and Applications - Proceedings of the European Safety and Reliability Conference, ESREL 2014, pp. 883–888. CRC Press, Balkema (2014)
6. Vogt, H.: Efficient object identification with passive RFID tags. In: Pervasive Computing. Lecture Notes in Computer Science, vol. 2414, pp. 98–113. Springer (2002)
7. Alemdar, H., Durmus, Y., Ersoy, C.: Wireless healthcare monitoring with RFID-enhanced video sensor networks. Int. J. Distrib. Sens. Netw. **2010**, 290–300 (2010)
8. Dong-Liang, W., Wing, W., Ng., Y., Chan, P.: Access control by RFID and face recognition based on neural network. In: IEEE Xplore Digital Library. http://ieeexplore.ieee.org/abstract/document/5580558/authors. Access date Mar 2017
9. Patel, U.A., Priya, S.: Development of a student attendance management system using RFID and face recognition: a review. Int. J. Adv. Res. Comput. Sci. Manage. Stud. IJARCSMS **2** (8), 109–119 (2014)
10. Meng, X., Song, Z., Li, X.: RFID-based security authentication system based on a novel face-recognition structure. In: IEEE Xplore Digital Library. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5571716. Access date Mar 2017
11. http://www.codeproject.com/Articles/239849/Multiple-face-detection-and-recognition-in-real. Access date Feb 2017
12. http://sourceforge.net/projects/emgucv/files. Access date Feb 2017

Aspect-Oriented Management of Service Requests for Assurance of High Performance and Dependability

Paweł Lubomski[1], Paweł Pszczoliński[1](✉), and Henryk Krawczyk[2]

[1] IT Services Centre, Gdańsk University of Technology, Gdańsk, Poland
{lubomski,pszczola}@pg.gda.pl
[2] Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Gdańsk, Poland
hkrawk@eti.pg.gda.pl

Abstract. A new approach to service requests management in case of insufficient hardware resources is proposed. It is based on wide aspects of requests analysis and it assures reliable and fast access to priority services. Requests are analyzed for, among others, time of occurrence, category of user who made the request, type of service, current system load and hardware utilization. Deterministic but dynamic rules help to manage system load very effectively, especially in terms of dependability and reliability. The proposed solution was tested on Gdańsk University of Technology central system, followed by the discussion of the results.

Keywords: Service requests management · Aspect · Dependability · Reliability · Load balancing

1 Introduction

Ensuring efficient and reliable access to IT services is a serious challenge. It becomes even more difficult when requests for different services fluctuate over the time. That is the reason why cloud computing is used so commonly. It allows to scale hardware resources in a very quick, relatively cheap, and efficient way. But what happens, when hardware resources are limited? Depending on organization where the system is used, load can vary in a daily, weekly, monthly or even yearly cycles. A university is a very good example at this point. Administrative staff and university lecturers use their university system during working hours. Students, on the other hand, prefer evening sessions. In addition, depending on the moment of academic year, also the need for various IT services is different. For example, reading lecture plans is heavily used at the beginning of the semester, but after a few days the number of requests for this service drops drastically. Moreover, it is possible to determine the type of each service – whether reading or writing the data dominates. An example of a university shows that the allocation of resources for specific IT services should not be constant over time but dynamically adjusted. The problem to solve is to develop such a method of service requests management which in case of insufficient hardware resources will provide

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_26

reliable and fast access to priority services. Of course, you can try to resolve this problem using different types of load balancing but it does not give you the possibility to prioritize requests, especially when hardware resources are limited. As a result of a simple load balancing, we can achieve the consumption of all available resources and, as a result, the denial of service very quickly. The aim of an effective solution is to use multiple nodes and to handle service requests dynamically depending on the aspect in which they are made. In this approach the critical services will be available regardless of the load at all times.

2 Motivation and Related Work

The problem of proper and effective load balancing has been under intensive research for years. Load balancing algorithms can be divided into two categories. There are static algorithms commonly used such as Round Robin and Weighted Round Robin. On the other hand, last connection as well as weighted last connection are dynamic algorithms commonly used. There are also works on other issues of cloud or distributed computing, e.g. cost-optimal scheduling on clouds [1], load balancing for distributed multi-agent computing [2], agent-based load balancing in cloud [3], communication-aware load balancing for parallel applications on clusters [4]. Also, there are approaches that apply genetic algorithms to dynamic load balancing [5]. Latest approaches focus on dynamic type-related task grouping on the same nodes [6] or weighted last connection algorithms with forecasting mechanisms [7]. There is also some research related to performance overhead while using virtualization in cloud [8]. Some other research was done focusing on type of communication used in resource allocation inside cluster and on load balancing – blocking or non-blocking connection, especially in the message-oriented model [9]. It is very important when clusters and data are located in different data centres spread all over the world. That impacts on availability very much.

Our aim is to focus not only on availability but also on dependability and reliability of services, where proper prioritization based on wide aspect of requests analysis takes place. It is very important when there are not enough hardware resources to process all requests at a time. This work is the continuation of our research on context analysis for better security and dependability of distributed internet systems [10].

3 Proposed Solution

In contrast to a simple load balancing, management of requests based on an aspect of their calls does not allocate resources evenly. Aspect-oriented management of requests means that during the request realization there is not only services invocation but also an additional functionality. This additional functionality includes reading the request attributes and then, on the basis of them as well as the configured rules, deciding which service node should be involved. This additional action is entirely separated from the main request realization. All requests should be analyzed this way. This is a perfect example of aspect-oriented programming. By using information about who, from

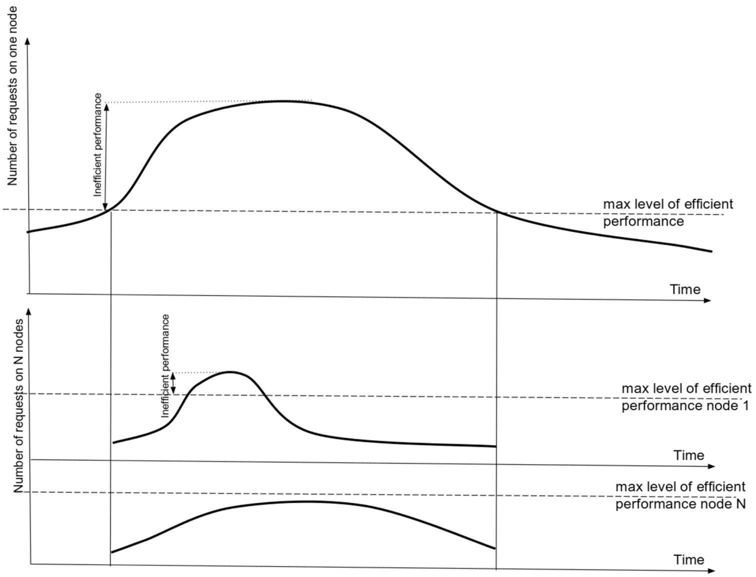

Number of requests on one node
Inefficient performance
max level of efficient performance
Time
Number of requests on N nodes
Inefficient performance
max level of efficient performance node 1
Time
max level of efficient performance node N
Time

Fig. 1. Management of service requests in terms of performance improvement with limited hardware resources

where, and when invoke request to the service appears we are able to allocate resources (often very limited) to meet changing demands and at the same time provide the resources needed for critical services dependability. Figure 1 illustrates schematically the definition of the problem and its solution.

System work on a single node during the biggest load is inefficient very often and many requests are serviced in sub-optimal time. Detailed statistics from our case are presented in Sect. 5. Adding more nodes with appropriate management of requests, although they did not allow all services to maintain optimal execution time (node 1), allows the selected priority services to work optimally and without any delay (node N) during the increased load time period. If you have resources to ensure optimal performance of all services during the biggest load, a simple load balancing is good enough to ensure efficient performance. Dynamic management of requests on the basis of modifiable configuration is a necessary solution when it is not possible to ensure sufficient resources for all services and while priority services must be provided with high performance and reliability during the whole period of increased load. In single node configuration during the peak load snow ball effect appears, which means that more and more services response in elongate time because they are waiting for access to resources. In such situation whole system work with poor performance. When we can divide load between many nodes and decide that on some of them only priority services will work those services will not wait for resources. Even in peak load snowball effect will not appear on nodes where demand for resources is less then available.

Of course, the proper system architecture is important to support aspect-oriented management of requests. A perfect solution is to introduce an explicit separation between the user interface layer and the services layer. It is assumed that the user interface layer (web browser with running applications on the client side and the network connection between the browser and the server) does not bring noticeable performance drop during the request handling. Comparing with the time of the request handling in the services layer, the time of transmission and request handling in UI layer can be omitted. The discussed model of system architecture is presented at Fig. 2. This architecture is used in the increasingly popular microservices [11].

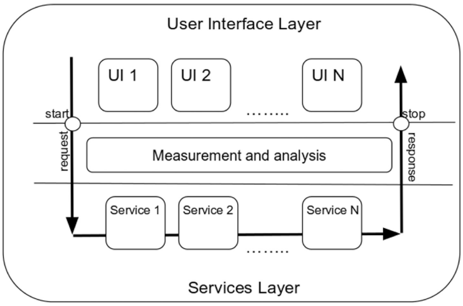


Fig. 2. Model of system architecture with request analysis between UI layer and services layer

Building a system that uses architecture of independent services and applications assures flexibility. The problem of providing communication between services and UI application requires an additional usage of REST and JSON protocol at a small effort. The explicit separation of the UI layer from the business services layer provides a number of advantages:

- less impact of errors for overall work of the system - by separating the requests, a part of the system can be in error state while the remaining part is able to work properly,
- scalability – you can easily deploy services that require more resources on machines with greater computing power,
- the ability to customize system architecture to the realities of the organization - this is important when the organization is large and has a complicated structure. It also allows you to deploy selected applications and services for specific groups of users to ensure the improvement of access control,
- the possibility of a partial modernization of the system – this is especially important for large systems where frequent exchange of technology is impossible. With independent services it is easier to upgrade some parts of the system. It is necessary to ensure compatibility at the API level only.

As business services have only programming API and they do not have a user interface, developers are forced to write unit tests because otherwise they are unable to test what they have produced. The key to the implementation of high quality software is writing automated tests. Unfortunately, when quick results are expected, this activity is often omitted in the first stage of the system production due to the additional time effort. However, when the work methodology forces developers to create tests, then they will guarantee high quality solution at the maintenance stage. That and all previous advantages make the architecture of independent services very functional and easily applicable, especially in systems with microservices architectures [12].

As for an aspect-oriented management of request deployment, the first thing to do is determining the system's characteristics. It means indicating time periods when the system is the most intensively used, when delays in access to services happen which result from inadequate hardware resources for the system's load. The next question is in which services the delay occurs and if it is caused by the massive use of services by users or the services' insufficiency. Another matter is if any services should be distinguished in terms of importance and priority. To answer all these questions you need to perform some suitable measurements. The best way to deal with that matter seems to be gathering traces of performed operations and their execution times on the service layer. Additionally, it is worth saving the aspect of request characterizing the selected call: which user invoked the request (what users' group he belongs to), which application the request was invoked from, when it was invoked and which service it was sent to. After collecting these data and analyzing them, we can determine which services consume the most resources, when, by whom and from what applications they are used. You need to compare the above information with the business environment of the system: who (which group of users) has to be provided with priority access, to what services and in which period of time this access must be provided. In practice, the most often it turns out that various services are not used extensively simultaneously and that miscellaneous groups of users need priority at different time. Such dynamics of the system usage enforces a similar dynamic in allocating resources.

The allocation of limited resources to services must not be constant but variable with the ability to adapt easily, depending on the aspect of the request. For this purpose, the requests management component must be introduced between the UI and the services. That component should be able to analyze the mentioned above aspects of a request. Then, depending on predefined rules, the request should be redirected to one of the N nodes. In this way, you can specify the nodes that will serve a selected group of users and requests to selected services. This allows you to prioritize periodically the selected requests by providing reliable access to critical services and functions of the system.

Figure 3 shows a formal diagram of request redirection to the nodes on the basis of the aspect of the request and the defined configuration. The ideal solution is defining the rules in the way which will ensure equal responsive access to all services for all users. However, with a large number of users and services and with limited resources it may be impossible. Then you have to choose which critical services should be available in which periods of time.

The rules should be under modification as long as we reach the satisfactory level of responsiveness of critical services. At the time of insufficient resources, low priority services may not be available but those with a high priority will be able to work properly. Figure 4 shows a life cycle of configuration. It will be usually a daily cycle, but it can also be adapted to another time quantum, depending on the needs.

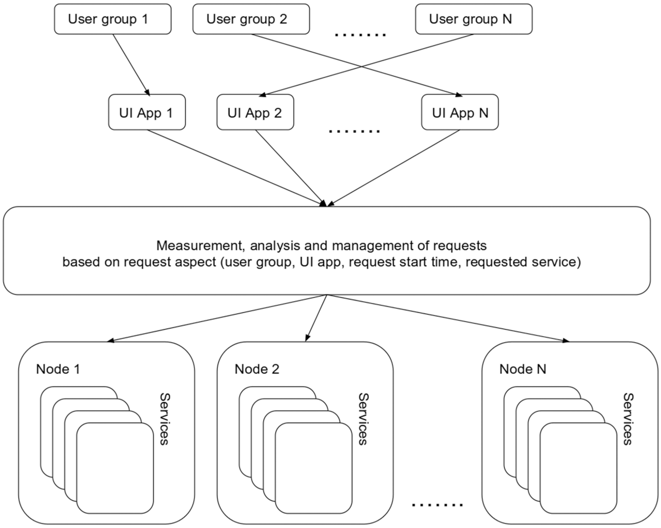

User group 1
User group 2
....
User group N
UI App 1
UI App 2
....
UI App N
Measurement, analysis and management of requests
based on request aspect (user group, UI app, request start time, requested service)
Node 1
Services
Node 2
Services
....
Node N
Services

Fig. 3. Diagram of requests redirection to the nodes on the basis of the aspect of the request

Continuous monitoring of requests and the collected data analysis allows you to customize the configuration of dynamic requests to the nodes allocation. This way you can achieve the highest possible efficiency as well as ensure the dependability of key services.

4 Verification Method – Case Study

The proposed solution was implemented in the system "Moja PG" at Gdańsk University of Technology. A component of request management was introduced between the portal (which is a collection of independent user applications) and services layer (in which business components are embedded). Another node was added on which all services are available in the same way as on the original node. The requests manager distributes requests from user applications into these two independent nodes.

Using exactly the same UNIX operating system and the same version of the JVM on both nodes provides the same threading, concurrency and parallelism mechanisms. Exactly the same system software was used before and after adding the request manager. Thus it is possible to compare the results of those two measurements without the need to consider the impact of the operating system or JVM.

Dynamic requests management solution was introduced in the production environment at the beginning of the academic year 2016/2017. It was a specific period of time because you had to ensure reliable access to services for 30,000 users within a few days. Students wanted to see their timetables, sign up for elective subjects or extend the validity of their electronic ID cards. At the same time employees had to issue the necessary certificates, give students their final grades and supervise the teaching process. Student actions during the working day got lower priority in the access to

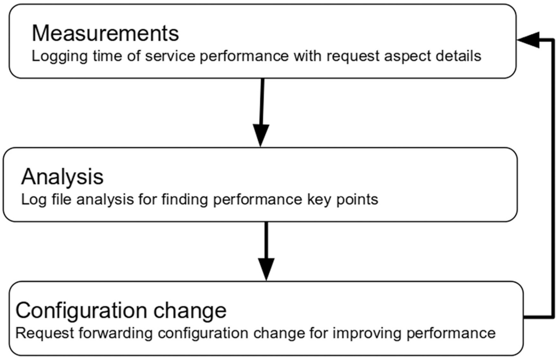

Measurements
Logging time of service performance with request aspect details
Analysis
Log file analysis for finding performance key points
Configuration change
Request forwarding configuration change for improving performance

Fig. 4. Life cycle of requests redirection configuration

resources on the basis of the aspect of the request. However, full access to resources was granted in the evening when university employees did not use the system.

Introducing such a separation, the average execution time was reduced. There was no noticeable drop in performance during the first days of the academic year, which occurred in the previous years. Services were available, there was no moment in which the system was overloaded. Thanks to the dynamic request management and despite a sudden increase in the number of system users, there was no snow ball effect. This effect means lengthening the service response time as a result of taking over more and more resources by other services. Reliable access to critical services was guaranteed all the time because they were invoked on a separate node. The change from a traditional balanced load distribution to dynamic request management helped to maintain full reliability of critical system functions.

The process of determining the rules for the separation of requests between nodes was discrete and was repeated once a day. Analysis was performed automatically on the basis of log files in which service requests were saved along with their execution times. As a result, we were able to calculate the following statistics:

- average time of service execution divided into groups of users,
- the total time of the service execution during the day divided into groups of users,
- the number of requests to the service divided into groups of users.

On the basis of those statistics it is easy to determine which services are the most often invoked, by whom, and which occupy resources for the longest time. Of course, changing the configuration directly affects the performance of the execution of services, so you need to monitor regularly the statistics to ensure that the rules are properly applied. Besides, changes in the way of using the system make a continuous control essential, too.

5 Measurements Results

Measurements of system “Moja PG” in the production environment were made on log files generated by the system. A sample log file part is shown below:

```
2016-09-26 00:00:05,170 [StudentManagerBean t-146] START: getObjectById
2016-09-26 00:00:05,205 [StudentManagerBean t-146] STOP getObjectById: 34ms
2016-09-26 00:00:05,222 [StudentManagerBean t-18] START: getCardBySubject
2016-09-26 00:00:05,662 [InventionsManagerBean -83] START: searchInventions
2016-09-26 00:00:05,794 [InventionsManagerBean t-83] STOP searchInventions: 132ms
2016-09-26 00:00:05,809 [InventionsManagerBean t-157] START: getInventorsByInventionId
2016-09-26 00:00:05,822 [InventionsManagerBean t-157] STOP getInventorsByInventionId:12ms
2016-09-26 00:00:05,835 [InventionsManagerBean t-146] START: getAdministrativeUnitsByInvenId
2016-09-26 00:00:05,838 [InventionsManagerBean t-146] STOP getAdministrativeUnitsByInvenId: 2ms
2016-09-26 00:00:05,844 [StudentManagerBean t-18] STOP getCardBySubject: 621ms
```

Some part of the following lines content logged by the system had to be hidden for security reasons. Pairing log lines talking about the beginning of the service request processing (START) and its completion (STOP) was performed on the basis of the task ID (e.g. “t-146”). Each STOP line contains also execution time in milliseconds. Services are executed asynchronously, so log lines appear in a file in a random order, too. In each line there is also included the timestamp, the name of the package of services, e.g. StudentManagerBean or InventionsManagerBean, and the name of the proper service.

The first log analysis checked only correctly completed invocation of services and summed up their execution times. Then, on the basis of the total time in one-day periods, a ranking list was created showing which services performed the longest. The ranking presents services that are invoked very frequently or which are invoked rarely but with a long execution time. Therefore, it is also worth analyzing the average time of service execution. First, the measurements were performed on a loaded production system without the additional node. Then, the services which occupied the first node resources for the longest time were redirected to the second node. After 24 h log file analysis was repeated, this time on both nodes. The collected results are shown in Tables 1, 2 and 3 and in Figs. 5 and 6.

The charts on Figs. 5 and 6 show the average execution times of selected services at the beginning of the academic year in 2015 and 2016. In 2015 “Moja PG” system worked only on a single node without aspect-oriented management of requests. In 2016 an additional node was used and service requests were distributed between nodes on the basis of their invocation aspects. The noticeable fact is the reduced service execution time (the longest service executions in 2016 lasted 765 ms and in 2015 – 1907 ms). In addition, on the graph of 2015 you can notice a significant increase in services execution time on 2015-10-02 – all students started to use the system intensively at the beginning of the academic year. Then a snowball effect occurred – an extension of services execution time because the user load caused that each invoked service was executing even longer. In the chart of the year 2016 there was an extension of services execution time on 2016-09-30, when the students started to use the system

Table 1. The results of service performance measurement in 2016

Year 2016				
Service name	Max execution time [ms]	Avg. execution time without load [ms]	Avg. execution time with load [ms]	The increase in execution time between the loaded and no-loaded system [%]
getInvoiceSummary II	108.00	90.50	96.13	6.22
getMyCourseTree I	317.00	200.50	253.80	26.58
getMyGeneralInfo II	437.00	348.50	368.07	5.61
getMyPersonData II	15.00	14.03	14.33	2.16
getMyStudyProcessCard II	700.00	581.50	628.20	8.03
getObjectById I	765.00	547.50	638.53	16.63
searchFullCourseEctsByCourseId II	20.00	15.00	15.20	1.33
searchMyStudentMessage I	692.00	523.00	578.33	10.58

Table 2. The results of service performance measurement in 2015

Year 2015				
Service name	Max execution time [ms]	Avg. execution time without load [ms]	Avg. execution time with load [ms]	The increase in execution time between the loaded and no-loaded system [%]
getInvoiceSummary	172.00	71.17	94.67	33.02
getMyCourseTree	370.00	241.67	324.67	34.34
getMyGeneralInfo	1286.00	258.00	455.00	76.36
getMyPersonData	69.00	17.67	32.44	83.65
getMyStudyProcessCard	1907.00	1060.67	1670.78	57.52
getObjectById	1434.00	966.33	1270.56	31.48
searchFullCourseEctsByCourseId	157.00	43.33	56.11	29.49
searchMyStudentMessage	1078.00	552.67	820.22	48.41

intensively. However, thanks to using two nodes, there were enough resources to prevent any snowball effect and the duration of execution time of services did not exceed 800 ms.

Measurements from 2016 presented in Table 1 clearly show that the priority services running on the second node had a single-digit increase in execution time under load (from 1.33% to 8.03%), while services operating in the more loaded first node had

Table 3. The results of comparative measurements of service performance in 2015 and 2016

Service name	Increase in max execution time between 2016 and 2015 [%]	Increase in avg. execution time without load between 2016 and 2015 [%]	Increase in avg. execution time with load between 2016 and 2015 [%]
getInvoiceSummary	−37.21	−10.54	−33.55
getMyCourseTree	−14.32	−17.03	−21.83
getMyGeneralInfo	−66.02	−4.52	−19.11
getMyPersonData	−78.26	−20.58	−55.82
getMyStudyProcessCard	−63.29	−45.18	−62.40
getObjectById	−46.65	−43.34	−49.74
searchFullCourseEctsByCourseId	−87.26	−65.38	−72.91
searchMyStudentMessage	−35.81	−5.37	−29.49

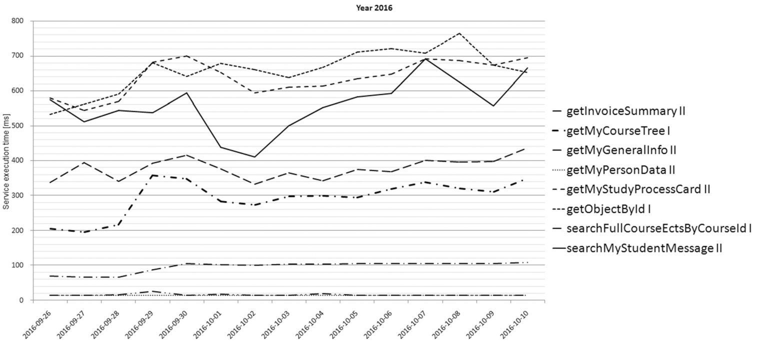


Fig. 5. Chart of the average service execution time on selected days of the year 2016

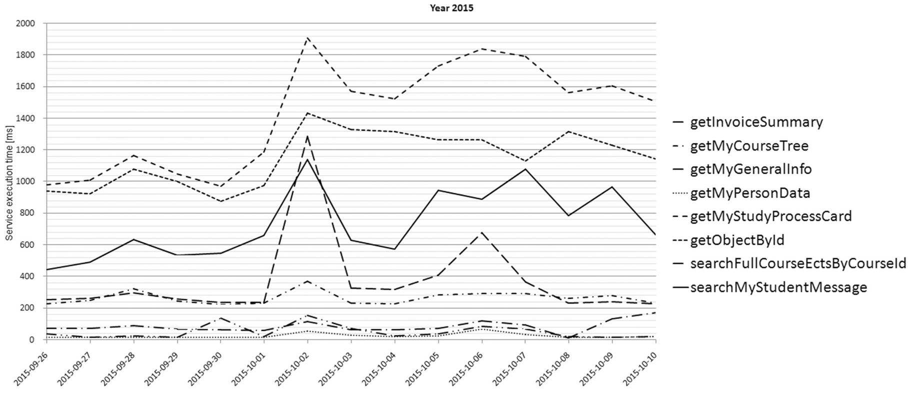


Fig. 6. Chart of the average service execution time on selected days of the year 2015

an increase in execution time under load in the range from 10.58% to 26.58%. On the basis of those results it can be stated that the critical services running on a node II worked stable and reliably, despite the system maximum load of 30 thousand users. If the load had been distributed equally between the two nodes, some of the resources would have been used by the services with a lower priority, and consequently the priority services execution times would have been extended, affecting negatively on the dependability of key system functionality. Table 3 shows compiled results of the comparison of services execution time between selected time periods in 2015 and 2016.

Without load the percentage differences in execution time ranged from 4.52% to 65.38%, which means that the system running without load on one node selected services (getMyGeneralInfo, searchMyStudentMessage or getInvoiceSummary) handled at the optimal time. This is why differences are so small (from 4.52% to 10.54%), while the execution time of remaining services even without load were reduced from 17.03% to 65.38%. This indicates that one node even without load did not have enough resources for optimal work. The comparative analysis of the maximum service execution times between unloaded system and system under load shows an increase of execution time from 14.32% up to 87.26%. The services of higher priority that were transferred to the second node (II) were being executed shorter from 37.21% to 87.26%, while those of lower priority which were executed on more loaded first node (I) showed a shorter duration of execution time in the range from 14.32% to 46.65%. Summing up the results of the measurements, it can be stated that with the selection of priority services and redirection of their execution on less loaded node a shorter execution time and reliable access, even at maximum load, were provided. Services with less priority which were running on the more loaded node also recorded a reduction of operating time (by distributing the execution of services on two nodes). However, it was not as significant as in the case of priority services. The measurements' results show that using an aspect-oriented management of service requests can assure dependable access to priority services which can be moved to a separated node. Of course, when there are unlimited resources, a dynamic load-balancing is enough for making all services unfailing. However, when there are insufficient resources, the services should be prioritized and the most important ones should work properly regardless of the load. In the production environment we were able to achieve our main goals by adding another node and using an aspect-oriented management of service requests. In this way:

- the execution time of all services decreased because more resources were available,
- the execution time of priority services decreased significantly because they were invoked on a separated node,
- the most important goal – priority services guaranteed reliable performance regardless of the load.

Priority service are used by finite and well known group of users. We were able to estimate how much of resources were necessary for this group of users to work on priority services. During work day of this group of users required resources were granted to priority services which gave us confidence that they will work with high performance and dependability. When priority services were not used, thanks to dynamic configuration free resources were used by other services which needed them.

Management of service request allows to react to a changing load and assure necessary resources for the chosen services. When high performance and dependability cannot be assured for all services because of the lack of resources, then a configurable management of service requests is a good solution. Thanks to that the priority services can perform efficiently.

6 Summary

The proposed solution is based on the aspect-oriented management of services requests. It was introduced in the system “Moja PG” at Gdańsk University of Technology that supports 30,000 users. After adding an additional node, the services execution time was reduced, thanks to a dynamic management of requests. On the basis of the request aspects and configurable rules, a snowball effect was avoided in the time of the most intensive system use. Execution time of priority services got reduced by an average of more than 60%. In the case of services with lower priority, the execution time was reduced by more than 30%.

Summing up, when computing resources are limited, then appropriate arrangement of rules for requests management provides a stable and fast access to critical services, without the risk that less important services will lead to the seizure of resources. Of course, the rules must be constantly modified, based on the performance results in order to adapt them to the changing conditions of system usage. As a result of the need for regular performance analysis and customization of the rules used in the configuration, there arises a question if the system should be allowed to configure itself automatically. Some work on the mechanism of dynamic recommendations was started. At this stage we assumed that the system should suggest which rules will improve the efficiency of access to services. However, the administrator will configure these rules in the production environment manually. When the recommendation engine is refined, an ultimate goal is to implement a module which will automatically reconfigure the system due to changing user activities.

References

1. Van den Bossche, R., Vanmechelen, K., Broeckhove, J.: Cost-optimal scheduling in hybrid IaaS clouds for deadline constrained workloads. In: 2010 IEEE 3rd International Conference on Cloud Computing, pp. 228–235 (2010)
2. Metawei, M.A., Ghoneim, S.A., Haggag, S.M., Nassar, S.M.: Load balancing in distributed multi-agent computing systems. Ain Shams Eng. J. **3**(3), 237–249 (2012)
3. Gutierrez-Garcia, J.O., Ramirez-Nafarrate, A.: Agent-based load balancing in cloud data centers. Cluster Comput. **18**(3), 1041–1062 (2015)
4. Qin, X., Jiang, H., Manzanares, A., Ruan, X., Yin, S.: Communication-aware load balancing for parallel applications on clusters. IEEE Trans. Comput. **59**(1), 42–52 (2010)
5. Zomaya, A.Y., Teh, Y.-H.: Observations on using genetic algorithms for dynamic load-balancing. IEEE Trans. Parallel Distrib. Syst. **12**(9), 899–911 (2001)

6. Lin, C.-C., Chin, H.-H., Deng, D.-J.: Dynamic multiservice load balancing in cloud-based multimedia system. IEEE Syst. J. **8**(1), 225–234 (2014)
7. Ren, X., Lin, R., Zou, H.: A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast. In: 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, pp. 220–224 (2011)
8. Lubomski, P., Kalinowski, A., Krawczyk, H.: Multi-level virtualization and its impact on system performance in cloud computing. Commun. Comput. Inf. Sci. **608**, 247–259 (2016)
9. Zenon, C., Venkatesh, M., Shahrzad, A.: Availability and load balancing in cloud computing. In: International Conference on Computer and Software Modeling, IPCSIT, September 2011, vol. 14, pp. 134–140. IACSIT Press, Singapore (2011)
10. Lubomski, P., Krawczyk, H.: Clustering context items into user trust levels. Adv. Intell. Syst. Comput. **470**, 333–342 (2016)
11. Thones, J.: Microservices. IEEE Softw. **32**(1), 116 (2015)
12. Balalaie, A., Heydarnoori, A., Jamshidi, P.: Microservices architecture enables DevOps: migration to a cloud-native architecture. IEEE Softw. **33**(3), 42–52 (2016)

Process of Mobile Application Development from the Security Perspective

Aneta Majchrzycka and Aneta Poniszewska-Maranda(✉)

Institute of Information Technology,
Lodz University of Technology, Łódź, Poland
aneta.poniszewska-maranda@p.lodz.pl

Abstract. Recent tendencies in the field of application development indicate the more and more significant role which mobile applications fulfill in everyday live of the rapidly growing number of smartphone users. Development of mobile application fulfils more and more important role in the everyday lives of the visibly growing number of smartphone and tablet users especially from the pint of view of security aspects. Addressing these tendencies the paper presents the security development with the algorithm constituting the creation process of mobile applications to overcome the existing threats faced by the mobile application developers.

Keywords: Mobile application development · Mobile security · Data access security · Data storage and transfer security

1 Introduction

Currently there exists no standardized solution or methodology for developing the applications which would be ultimately threat-resistant. Together with the invention of newer and newer strategies to prevent the security breaches the more sophisticated threats emerge and new vulnerabilities are detected by the platform developers. While designing and building the mobile applications there will always be a tradeoff between functionality, optimization and security. However none of these areas should be omitted and treated with less importance than the others as only balanced combination of those three will enable to create the reliable and useful solutions [2, 3, 6].

The increase in mobile Internet traffic is mainly due to the increasing possibilities of mobile devices offered to the users. However, the intuitiveness and ease of use of Android applications make it comparably attractive despite its known drawbacks in the area of security. Any developer of mobile application for these platforms should be aware of the threats and vulnerabilities that each of them carries and should adjust the development strategies in such a way so that the optimal level of safety is assured especially when interaction with confidential or sensitive data is required [1, 4, 7].

The model called *Security Development Strategy (SDS)* [10, 11] was developed in the previous research works for building mobile applications so that they would be less vulnerable to external attacks and leaks of sensitive dat. The existing another approaches to mobile application security focus mainly on the transmission of sensitive data to external services, whereas using the SDS approach equal focus is put on all

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_27

aspects of the sensitive data management. Safe data transfer between mobile and external devices is undoubtedly a crucial link in the process of securing the applications, however not the only one.

The problem presented in the paper concerns the development aspects of mobile applications from the point of view of their security. It describes the general overview of algorithms constituting the creation process of mobile applications to overcome the existing threats faced by the mobile application developers. It proposes the design pattern with some detailed technique suggestions to improve the security of developed mobile applications.

The presented paper is structured as follows: Sect. 2 gives the outline of Security Development Model created for building secure mobile applications and the algorithm of mobile application development. Section 3 deals with the algorithm of first pillar of SDS, data storage security model. Section 4 presents the algorithm of second pillar of SDS, data access security model while Sect. 5 describes the algorithm of third pillar of SDS, the data transfer security model.

2 Security Development Model

The field of security of mobile applications requires the elevated attention because of the privacy issues of millions of users of smartphones and the lack of adequate solutions to assure the security of data. Having this in mind it is worth to think about more ways of how to reduce the risks connected with the mobile security in the context of the development process itself [5, 6, 8, 9]. One of the most crucial aspects which is the core of the proposed solution is to create a model which assures less probability of capturing the sensitive data by the malicious software or hackers [5, 6, 8, 9].

The idea of *Security Development Strategy* [10, 11] assumes that application should conform to predefined security standards embracing three main areas: *storage* (of sensitive data on the mobile device), *access* and *transfer* of sensitive data (Fig. 1). Conformation to the standards should be achieved by implementing threefold security pattern for each of the mentioned areas. The model specifies the assumptions on how to achieve a proper level of security in each field and provides necessary details on the implementation of mechanisms which will allow achieving desired safety effects.

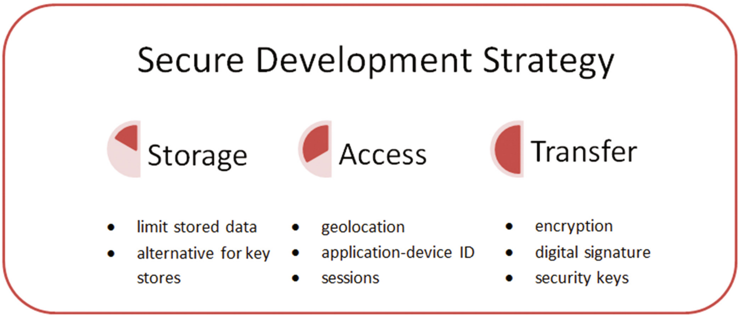


Fig. 1. Pillars of Security Development Strategy for mobile applications

The process of mobile application development from the security perspective constituting the *Security Development Strategy (SDS)* is regarded as an integral process during creation of any function of the mobile application.

The developers add security management features at the beginning of mobile application development and remember to add them every time they create a function handling sensitive data. Therefore the general structure of the process contains the following steps: Create project. Add security management. Create function. Handle security. Repeat for all relevant functions. Test security management. Improve security management. Publish application.

The algorithm of mobile application development from the security perspective contains the following steps (Fig. 2):

1. Define the scope of sensitive data, i.e. having the data structure of the application define which data will be considered as sensitive. For example in *Password Manager* application the sensitive data includes user credentials, user personal information, passwords stored within the application.
2. Define the *security level* for the application. The security level describes how the application should behave in case of security breaches. By default there are three levels of security levels:

   - *minimum security* – ignore breaches,
   - *medium security* – alert about the breaches, but do not lock the application,
   - *maximum security* – alert about the breaches and lock the application.

3. (optional) Specify the allowed locations for the application usage (e.g. applications used in stores, hotels, restaurants – never outside a certain range).
4. Create *SecurityManager* class through which all requests for sensitive data should be handled.
5. Create the function(s) of an application.
6. Specify in XML configuration the permissions for the function(s) which manipulate the defined scope of sensitive data.
7. Repeat steps 5 and 6 whenever a new function handling sensitive data is being created.
8. Perform security tests. The security tests should include all the security mechanisms from *SecurityManager* class which are used in the application. The scenarios of the security tests have to be created. They should embrace all the possible courses of actions for the given functionality. For example for login process embrace:

   - user gives valid credentials,
   - user gives invalid credentials one time,
   - user gives invalid credentials X number of times, after which the application should be locked,
   - the same login credentials are used from two remote locations – alert e-mail should be sent to the user.

9. If necessary implement improvements in the security configuration.
10. Repeat steps 8 and 9 until security tests results are satisfying.

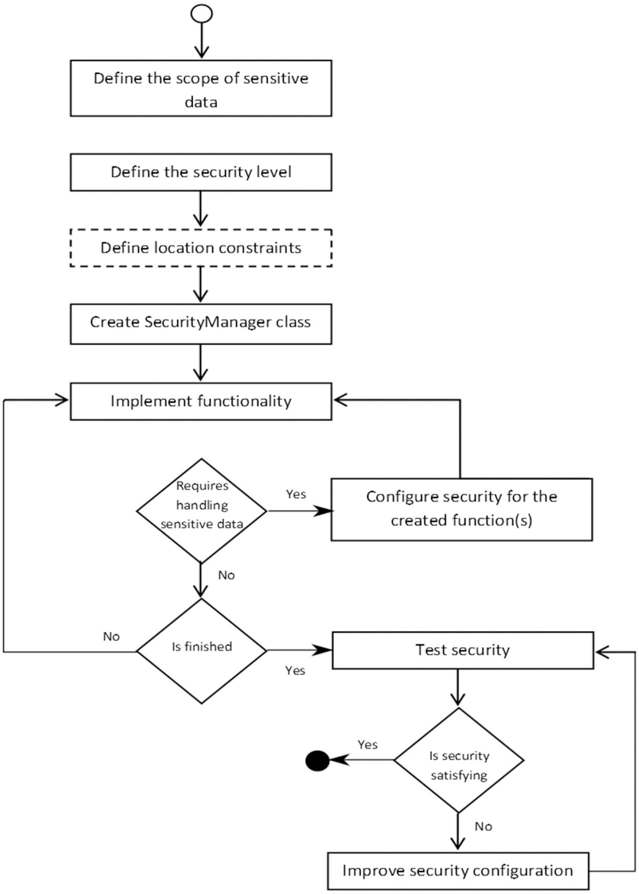

Define the scope of sensitive data
Define the security level
Define location constraints
Create SecurityManager class
Implement functionality
Requires handling sensitive data
Yes
Configure security for the created function(s)
No
No
Is finished
Yes
Test security
Yes
Is security satisfying
No
Improve security configuration

Fig. 2. Algorithm of security mobile application development

3 Data Storage Security Model

The first pillar of SDS storage concerns the client-side of the system i.e. the mobile application. The major assumptions of data storage pattern embrace sensitive data encryption, limitation and restricted access.

Mobile applications developer has two possibilities on where to store application data. He can choose external server where data will be stored in databases and special firewall mechanisms will block access to it. On the other hand some of the information which the application uses is necessary to be stored on the device. The first option seems to be a better solution as it eliminates the risk of losing data when the device is damaged. However, it requires a large amount of data traffic between the application and the server. In that light the second option comes in handy – it reduces the amount of data transfer. It seems to be less practical, as the data to be valid need to be updated. Moreover, the storage space of the device is also limited. Thus, the combination of both solutions comes from the need of keeping the data up-to-date and accessible by many devices at any time simultaneously giving the possibility to store a little number of crucial information on the device [11].

The storage mechanisms depend on the place of where the data is saved on the device. Two places for data storage which are also a potential risk points can be discussed for SDS: key-chain and device file system.

3.1 Key Stores

Mobile platform developers introduced special mechanisms for storing data in a secure manner – key stores. The assumption is that the applications can access only those key store values which they have privileges to. For jailbroken devices there is no problem in reading the key store values with the use of special dumping applications. This is the reason why the first assumption of SDS states that: *Sensitive data should never be stored as plain text, but they should always be encrypted and stored as such in key-chains and any other storage places.*

The other suggestion for storing sensitive data is to transfer the responsibility from key-chains which are known to store valuable information, especially remembered passwords, to the internal database of the application. This is expressed as the second rule for data storage: *Sensitive data could be stored within the application database files and encrypted using encryption keys stored in the external server databases to limit the risk of reading the data.* The algorithm for limitation of storage and secure storage constituting data storage security model in SDS is as follows:

1. Define the scope of data that the function which is currently being implemented requires.
2. Classify the data according to its sensitivity.
3. Specify the necessity to keep the data on the device; for example the registration process.
4. For each piece of data not required to be stored find another way on how user will be accessing it. The example on when the data does not require to be stored is the user personal information during the registration or login processes. This personal information will be downloaded from the server every time the user logs in to the application.
5. For each piece of data to be stored on the device.
    1. Create a separate keychain.
    2. Define a keychain entry.
    3. Store the name of the keychain entry in the XML configuration file. The XML configuration file was chosen as the most common approach to store the configuration information. It is widely used, understandable and most of the programming environments support the reading and writing to and from the XML files.
    4. Add the encrypted value to the created keychain entry. The choice of the encryption algorithm depends on the developer. By default the AES algorithm is being used.

3.2 File System

Sensitive data could be stored within the internal database as custom structures known only for developers of a specific application. This is not a common practice at the time. The SQLite data records are stored in a file on the device, thus it may seem a hazardous thing to store the passwords there. That is why credentials and simple character variables should be still stored with the use of key-chain stores, however in case of other structured data one should definitely use the internal database of the device.

The algorithm for function privileges specifies for each function in the configuration file which data it can have access to. It constitutes data storage security model in SDS as follows:

1. Define the scope of data which will be accessed by the function.
2. Implement function (write function body).
3. If the required data is qualified as sensitive go to step 4, else go to step 7 and finish.
4. If it was not done previously add reference to the *iSec Framework* [11] through which access to sensitive data is obtained. The *iSec Framework* provides an interface for implementing security mechanisms into the application. Having referenced it we are able to call all the functions which it provides and use them to build the security into our application.
5. Modify the chosen function and use the built-in *iSec Framework* [11] method to access sensitive data.
6. Add the function name to the XML configuration file.

4 Data Access Security Model

The second pillar of SDS strategy concerns the access to data. This comes from the fact that mobile applications need to communicate with external services and other applications. The major assumptions of this security area embrace three mechanisms which aim to enable identification of the user requesting access to application resources: geolocation, application-device ID and sessions.

To ensure the functionality of the second pillar of SDS strategy, the access control model approach for mobile applications was proposed [11]. It was named mobile Application-based Access Control (*mABAC*) model. The core part of mABAC approach essentially represents the basic elements of access control, such as subjects, objects and methods.

4.1 Geolocation

The geolocation became a common mechanism for locating devices and performing certain services which enable to display proper language or time zone for the user. Recently Google has used this mechanism to secure access to private user accounts. The Google procedure works in such a way that the location of the computer together with date and time of the registration and consequent logins to the account are registered and then used for validation. The location of the computer is obtained on the basis of the IP address of the computer.

A similar mechanism can be implemented for a mobile application. Having a unique identifier of the device, its location could be registered and sent to the server with every request issued by the application. It would serve as a frame of reference for future logins. In case a suspicious attempt would take place a message should be sent to the user. This would require a mobile phone number or email specification by the user or the application could notify itself about the attempt to login with the use of push notifications. The algorithm for geolocation constituting data access security model in SDS is as follows:

1. Implement the login mechanism provided with the *iSec Framework* [11].
2. Implement the request handling mechanism of *iSec Framework* [11].
3. If the function requires the sending a request to the server go to step 4, else go to step 6.
4. Obtain current user's location coordinates.
5. Encrypt the coordinates and add them to the request.
6. Send the request.
7. On the server side resolve the coordinates based on incoming IP address.
8. Compare them with the coordinates found in the message body.
9. If the coordinates match, go to step 10, else go to step 13.
10. Check the history of previous user's locations.
11. If the location does not vary significantly from previous ones go to step 13, else go to step 14.
12. Allow data access and finish the algorithm (omit the steps 14 and 15).
13. Inform the user about the possible risk of violation.
14. According to the users response, block or allow access.

4.2 Device Unique Identifier

Every device is assigned a unique identifier which enables to distinguish it from the others. The transmission of a unique identifier can somehow violate the privacy as it could be captured by unwanted parties. Nevertheless, *Universal Device Identifiers (UDIDs)* seem to be the most reliable way which can be used to identify the requests for the data sent to the server. Together with the geolocation coordinates they can prevent external invalid requests to be served.

In order to conform to Apple standards it is best not to use the default device UDID, but it does not change the fact that we do not want to resign from the possibility of identifying a specific device. One has to remember that many users may log in to the application from the same mobile appliance. This is the reason why SDS suggests that a custom unique identifier was generated for the device during the installation phase. This unique identifier will be referred to as an *Application-Device Identifier (ADID)* and it will be characteristic for a single application only. That is why it will not be possible to use it for any other purposes than for the application in subject.

Application ID is stored in the configuration file of the server and the configuration of the application (*XML config file*). UDID is transmitted during the first application use and stored in the database (encrypted). The ADID is used to identify the specific device which uses the application. It enables to identify the device, the possible users and the application itself. The ADID enables to avoid the necessity to send the actual UDID of the device which by some is considered as privacy leak. The algorithm for ADID constituting data access security model in SDS is as follows:

1. Obtain application ID (AID) from the configuration file (it is installed together with the application).
2. Obtain device UDID.
3. Take 24 random characters from the UDID together with their indices in the UDID string.
4. Combine the obtained data into a single string consisting of an AID and additional 48 characters – one character from the UDID and its index, next character from UDID and its index, etc.
5. Encrypt it with the 3DES algorithm.
6. Send ADID together with every request to the server.
7. If the ADID (connected with a specific user and location) does not exist and the device requires access to other function than registration goes to step 8, else goes to step 9.
8. Display information about violation to the user.
9. Allow access.

5 Data Transfer Security Model

Data transfer pattern refers to all mechanisms which involve the exchange of data between the mobile application and the external services. These mechanisms should incorporate in their action flow the additional security procedures – data encryption, the use of security keys and the verification of the requests integrity [11].

Data transfer is the weakest link in the entire process of the mobile application development. It comes from the fact that the requests are travelling over the Internet in an unprotected space and they are prone to a special kind of attacks called the “man in the middle”. This attack means that between the mobile device and server application a third-party may be listening and waiting for the exchange of information.

5.1 Digital Signatures

As far as the integrity of the data is concerned it is a good and common practice to use the digital signatures. They enable to check whether the message received is exactly the same as the message sent and if it was not modified on the way to the receiver. The digital signature mechanism relies on the encryption which should assure the authentication. The *Digital Signature Standard (DSS)* relies on a private key algorithm. Only the owner of the message knows the private key and the public key is known.

It is however possible to use other encryption techniques and implement them instead. The algorithm for request integrity constituting data transfer security model in SDS is as follows:

1. Prepare the request.
2. Get current *timestamp*.
3. Hash the request with the current *timestamp*.
4. Add hashed string and encrypted *timestamp* to the request payload.
5. Send request to the server.
6. Resolve the hashed string with the *timestamp*.
7. If the string and the request body match go to step 8, else go to step 9.
8. Proceed with handling the request.
9. Display information about the lack of integrity of the message.

5.2 Security Keys

The last suggested mechanism represents security keys for digital signatures. Security keys introduced by SDS are 128-bit strings which should be sent to the server before the request for sensitive data. The security key can be unified for the entire application and independent from the device. When transferred it should also be encoded with chosen cryptographic algorithm. Security keys are used to sign the request and they allow verifying whether there was no violation of this request on its way from the user to the server. They contain the information about the ADID as which enables to identify the device from which the request was sent. They also incorporate date information which can protect from brute force attacks. The algorithm for security keys constituting data transfer security model in SDS is as follows:

1. Get the ADID of the application.
2. Get the current UTC *timestamp* (as Unix timestamp).
3. Combine the last 8 character from ADID and the date into one string.
4. Hash the string with ADID using SHA-3algorithm.
5. Add the security key to the request payload.
6. Send the request to the server.
7. The web service function on the server should create the appropriate security based on obtained *timestamp* and ADID stored on the server.
8. If the obtained security key and web service-generated key are the same, allow access. Otherwise inform the user about the violation.

6 Conclusions

The complexity of the ecosystem leads to the complexity of the mobile application development. Mobile applications have found to contain potentially malicious behaviors. The paper proposes a development process or even a design patter with detailed technique suggestions to improve the security of the applications.

This paper aims to address the problem on balancing the functionality, optimization and security of the mobile applications and proposes to take all aspects of sensitive data management during the application creation process into consideration. In specifically, it defines the security models of three data security aspects (i.e. data storage, data access and data transfer) and specifies the guidelines in the models to add security management features for each function which handles sensitive data. Moreover, the presented work describes a set of algorithms to support the development of secure mobile applications for all the three data security aspects.

In addition, a prototype framework is developed to implement the proposed model. The implementation of the ready-to-use classes and methods which would assure security of any mobile applications seems to suit the current needs for simultaneous time-efficiency and safety. That is why the prototype framework *iSec* was developed. Its main components correspond to three pillars of the security model presented in the previous sections: storage, access and transfer [10].

The further research on data storage model should be conducted in order to provide an easy to use way for developers to securely store indispensable data directly on the device. The possibilities include especially the data access control mechanisms.

References

1. Porter Felt, A., Finifter, M., Chin, E., Hanna, S., Wagner, D.: A survey of mobile malware in the wild. In: Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, pp. 3–14 (2011)
2. Souppaya, M.P., Scarfone, K.A.: Guidelines for Managing the Security of Mobile Devices in the Enterprise. NIST (2013)
3. Apple, iOS Security (2014). images.apple.com/ipad/business/docs/iOS_Security_Feb14.pdf
4. Zhou, Y., Jiang, X.: Dissecting android malware: characterization and evolution, In: Proceedings of the 33rd IEEE Symposium on Security and Privacy (2012)
5. Benedict, C.: Under the Hood: Reversing Android Applications. Infosec Institute (2012)
6. Enck, W., Octeau, D., McDaniel P., Chaudhuri S.: A study of android application security. In: Proceedings of the 20th USENIX Security Symposium (2011)
7. Fitzgerald, W.M., Neville, U., Foley, S.N.: MASON: mobile autonomic security for network access controls. Inf. Secur. Appl. **18**(1), 14–29 (2013)
8. Alhamed, M., Amir, K., Omari, M., Le, W.: Comparing Privacy Control Methods for Smartphone Platforms, Engineering of Mobile-Enabled Systems (2013)
9. Michalska, A., Poniszewska-Maranda, A.: Security risks and their prevention capabilities in mobile application development. Inf. Syst. Manag. **4**(2), 123–134 (2015). WULS Press
10. Michalska, A., Poniszewska-Maranda, A.: Secure development model for mobile applications. Bull. Pol. Acad. Sci. Tech. Sci. **64**(3), 495–503 (2016)
11. Poniszewska-Maranda, A., Majchrzycka, A.: Access control approach in development of mobile applications. In: Younas, M. et al. (eds.) Mobile Web and Intelligent Information Systems. LNCS, vol. 9847, pp. 149–162 (2016)

Managing and Enhancing Performance Benchmarks

Jakub Maleszewski and Janusz Sosnowski(✉)

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
jmalesze@mion.elka.pw.edu.pl,
J.Sosnowski@ii.pw.edu.pl

Abstract. The paper deals with the problem of computer performance evaluation basing on program benchmarks. We outline the drawbacks of existing techniques and propose a significant enhancement using available performance counters. This new approach needed development of supplementary tools to control benchmarking processes and combining them with various measurements. In particular, this resulted in some extensions of the open source program Phoronix Test Suite. The usefulness of our approach have been verified in many experiments taking into account various benchmarks and system configurations. We present the measurement methodology and interpretation of the obtained results, which confirmed their practical significance.

Keywords: Benchmark programs · Performance sensors · Performance evaluation · Test suite platform

1 Introduction

The technological progress in computer systems has a high impact on various utility features. In particular, it results in improvement of performance, dependability, power consumption, etc. An important issue is evaluation of these properties. For this purpose various program benchmarks have been developed. They can be targeted at different systems and aspects, and developed by commercial or scientific/industrial organizations. This topic is being discussed for many years in relation to appearing technologies ([5, 6, 8], conferences on benchmarking, e.g. [9, 17]). The proposed sets of benchmark programs are sometimes considered as controversial. Especially, in the case of superficially defined properties and technical documentation. Another issue is the problem of controlling and performing measurements. Hence, various measuring techniques and supplementary tools have been proposed [7, 14, 20, 22]. In our opinion classical benchmarking techniques can be significantly improved by combining them with performance counters.

The problem of system monitoring using performance counters have been addressed in many publications within our Institute (e.g. [11–13]), so we have gained good experience and developed some dedicated tools. Performing various experiments we have found the need of deeper studies taking into account different system configurations and their impact on the overall performance. This resulted in combining

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_28

benchmarking techniques with performance counters (neglected in the literature). This new approach needed developing some supplementary tools which have been integrated with a comprehensive testing and benchmarking platform [22].

In Sect. 2 we characterize benchmark programs and outline the problem of their analysis. Section 3 describes the developed methodology of enhanced measurements including performance counters. Section 4 presents and explains results obtained for a sample of experiments with different benchmarks and system configurations. Final conclusions are comprised in Sect. 5.

2 Benchmark Programs

In the process of evaluation various features of computer systems an important issue is selection of appropriate program benchmarks. In general, we can distinguish synthetic and application benchmarks. Synthetic benchmarks are based on specially developed programs targeted at imposing specific workload on the selected system components and their interaction [5, 6]. Application benchmarks are composed from real programs and measure overall system performance in relevance to the used application. In practice, it is reasonable to mix these two approaches. Selecting or designing benchmarks we should take into account workload specificity, e.g. its dynamics [6] and system domain (embedded systems [8], cloud environment [4], mobile systems [3], etc.). A static workload is one in which a certain amount of work is fixed and arrives at once. The dynamic workload is characterized not only by the workload items but also by a work arrival process.

Using benchmarks we should be conscious of the evaluation goals. In the stage of system development we are interested in optimizing system components so using various benchmarks we can check the performance of various solutions, the impact of specific components on the whole system behaviour, perform design trade-offs, etc. This goal can be further extended for system users in relevance to the possibility of changing system configurations, e.g. compiler optimizations, caching and buffering policies, memory organization and management, etc. [6]. Application designers can be interested not only in speed (performance of algorithms and platforms) but also in some high level user defined properties, e.g. the effectiveness of recognizing traffic signs [19] or other visual objects [3, 15]. Taking into account green computing paradigm and locally powered systems (Li-ion or solar batteries) power consumption becomes an important performance feature.

Another issue is considering dynamic changes (during system exploitation) of workload, memory or network usage, etc. Here, we can examine the impact of additional resources (even machines) on the performance (scale up and speedup elastic issues) [4]. On the other side it is reasonable to check the performance decrease in function of increased load. Developing subsequent system versions we face the problem of checking whether they still satisfy quality goals. This can be verified with so called regression benchmarks [20, 21] which simulate different usage patterns targeted at specificity of the tested systems. Before deployment of a new system it is reasonable to perform comparative performance evaluations [2].

Depending upon the goal of benchmarking and the target tested environment various benchmarks have been developed as open source or commercial ones. In practice they are targeted at some specific system platforms or issues, e.g. embedded systems (MiBench [8]), CPU architecture and processing kernels (e.g. SPEC CPU, linpack [6, 9, 14]), operating system scheduling, graphical processor (Unigine Valley), I/O subsystem (Flexible I/O Tester, HD Tune), memory usage and locality of references, data base, parallel systems ([1, 24]), transaction processing (TPC [16, 25]), PC and workstations; web, mail, system file server, disks (GNOME Disks, HD Tune), database (*pgbench*). In addition, we have a lot of popular application benchmarks e.g. *gzip* (file compression), *x264* (video coder in format H.264/MPEG-r AVC), C-Ray (image generation based on ray tracing), data centric and parallel processing applications. Here, we can include also real programs, e.g. word processing software, CAD software and other user's application software.

In most studies on benchmarking overall performance of the system is expressed by a single metric [1, 5, 6]. In the case of a single program typically the performance is given by the execution time, throughput or speedup in relation to some basic system. For benchmarks comprising different programs targeted at different system components or workloads the result aggregation is more complex. We can use here arithmetic mean (e.g. for time based metrics), harmonic mean (for rate based metrics, e.g. cache miss rate). In the case of benchmarks targeted at different features weighted means should be used. In many cases the aggregated metrics are not satisfactory, moreover they can be difficult for interpretation. Hence, we propose to enhance measurements with performance counters or other internal variables used as some probes (e.g. ratio of invoking specific program libraries).

Performance measurements allow designers and users to tune and configure systems, optimize application algorithms and codes, identify bottlenecks, match system features with workload characteristics. In most cases performance is correlated with the speed or system throughput. Recently, we are also interested in power consumption and dependability features. Hence, special benchmarks are developed for these purposes. Dependability benchmarking is intended to characterize system behaviour in the presence of faults [10]. Such benchmarks reveal weak points in the system, evaluate fault resistance of different solutions, etc. Dependability benchmarks are specified not only by the workload but also by fault load (internal and external).

Dedicated benchmarks usually comprise embedded measurement functions. Taking into account application benchmarks we have to perform some code instrumentation for measurements. Detailed (fined grained) analysis of various benchmarks is needed due to the fact that manufacturers provide and report benchmarks to show the best features of their products (bench marketing). Dealing with various programs as benchmarks it is important to automatize and manage measurements in testing processes. In particular, we have to log in real-time (while test is running) various performance sensors and present results. For this purpose we have enhanced popular and recognized benchmarking platform Phoronix Test Suite [22] with developed modules providing new capabilities using its extensible architecture (presented in Sect. 3).

3 Managing Performance Experiments

Performance evaluation needs running various benchmark programs with different input data for various configurations, Moreover, the performed tests should be repeated to get some statistics of results, which may differ due to various fluctuations in the evaluated system (e.g. processes related to operating system, program updates, memory usage organization, caching and buffering policies). Finally, the collected results should be presented in some numerical or visual form and analysed. Efficient handling of these processes in an automatic way needs a special test platform. An example of such platform is open source Phoronix Test Suite [22] (PTS). It is well accepted and easy to use by practitioners, providing wide range of configuration options not encountered in other benchmarking tools. Using this platform in our experiments we have found the need and possibility to enhance it with useful supplementary functions discussed in the sequel.

It is worth noting that classical benchmarks have limited capabilities of system configuration and no mechanism of integration with performance counters. They have embedded measurement functions which provide only parameters on basic usage features for the tested program, e.g. execution time, transmission speed from/to disk. On the other hand application benchmarks need additional code for the required measurements. These drawbacks are mitigated in a large extent in PTS platform which is written in PHP and works on many operating systems. It facilities managing complex tests with many benchmarks and their configurations in an automatic way. As an open source project it provides the capabilities of refinements, in particular a mechanism to extend its functionality with plugins, for example a built-in plugin capable of monitoring performance counters during test execution.

PTS is based on test profiles which are composed of configuration files specified in XML and various shell scripts describing actions needed to install tested applications (benchmarks). This allows us to automatize testing process starting from loading the application with supplementary files (e.g. libraries, files from Internet), performing measurements and presenting final results. Hundreds of test profiles from different categories are also provided with PTS, allowing quick and straightforward execution of tests to assess different aspects of system performance without specialistic knowledge. Test profiles are combined in test sets which can be executed sequentially within the same Phoronix session. Moreover, their results can be aggregated. The test set can comprise different benchmarks, and input data sets, including simulated environment interaction. Each test can be executed a specified number of times with optional additional repetitions performed till achieving a representative statistically significant features, e.g. standard deviation below certain threshold. PTS evaluates statistical properties of test results (e.g. standard deviation, minimal, maximal values). Moreover, PTS is integrated with bisection mechanisms of Git version control system, which allows to find code changes in subsequent versions related to performance reduction. Experiment results can be presented in some standard visual forms as well as in XML format, which is useful to add user defined analysis processes. The presented features make PTS a particularly useful tool to support benchmarking processes and minimize the threat of user-made mistakes affecting the result correctness.

Having analysed practically extensive functional features of PTS system we have found that it outperforms other open source benchmarking platforms. Hence, taking into account the gained experience with monitoring performance counters we have decided to extend the capabilities of PTS platform. The appropriate code extensions have been prepared by the first author. These extensions relate to three aspects: integration of performance counters into PTS, integration of PTS with GNU/Linux *cgroups* mechanism, improvement of MATISK (My Automated Test Infrastructure Setup Kit) module and plot generation.

Inclusion of measuring performance counters needed significant changes in PTS *system_monitor* module (*pts-core/modules/system_monitor.php*). In particular, the previous solution was restricted to monitoring a small set of sensors (performance counters) with static methods of *phodevi-sensor* interface. In consequence, it was not possible to create many instances of the same sensor type (e.g. to trace many CPU cores). Higher universality has been achieved by tuning *phodevi_sensor* into an abstract class (*pts-core/objects/phodevi-sensor.php*), which became a super class for all sensor classes. Further extension related to creating a list of sensors and verifications of correct selection of sensor parameters (e.g. disk identifier). Whenever possible, the available devices of a given type (e.g. disks, CPU cores) which can be monitored are automatically detected. Moreover, the previous sensor functions comprised many faults resulting from lack of source code updates needed to keep up with changes in standard GNU/Linux utilities used to obtain sensor data. The whole code has also been refactored. An important modification related also to improving time resolution of measurements by handling separately sensors with direct reads (e.g. temperature) and sensors introducing some measurement delays (e.g. CPU usage – counting time units within a specified time period, e.g. 1 s). The first group of sensors is handled with a common descendent process, the remaining by separate processes. In this way we avoid reduction of read frequencies while increasing the number of active sensors. Prior to modifications, all the measurements were performed sequentially. As a result, time resolution was constrained by the sum of mentioned sensor delays, what quickly became an issue with multiple instances of numerous sensors being monitored, especially during short tests.

An important issue is collecting performance data related strictly to the tested benchmark. Most tools monitor system resources by counting all parameter values which may cover also processes not related to the tested benchmarks. This problem has been resolved by integrating PTS with GNU/Linux *cgroups* mechanism [18]. It allows creating groups and assigning processes to them. Due to ability of resource accounting for individual groups, monitoring only processes and threads related to the tested benchmark is assured. When *cgroup* monitoring is enabled for the PTS test session, the new *control group* is created. Every process and thread spawned by the benchmark is then assigned to the newly created *cgroup*, which is destroyed after test process completion. Combined with the total resource usage metrics, the data obtained with *cgroup* accounting allows to estimate how the processes not related to the benchmark running during the test process impact the results, e.g. by consuming CPU time, which would be otherwise used by the benchmark itself. *Cgroups mechanism* provides *resource controllers*, which allow limiting usage of the specified types of resources (block devices, CPU usage, memory). This gives the possibility of testing the impact of resource availability on system performance.

It is quite useful to perform a set of tests based on contexts. The context describes configuration specific for invoking a specified test profile. In particular, this can be a set of compiler flags used in subsequent runs of the tested application or program version identifier. This capability simplifies automatic comparisons. Such functionality was a part of the built-in PTS MATISK module which disappeared in newer versions due to compatibility problems. The developed improved version of this module has been successfully integrated with PTS. Moreover, PTS has been combined with improved graphical result plotting (extended by introducing the capability of displaying simultaneously many test results in one view). All the extensions and improvements have been accepted by the project leader Michael Larabel and are included in the version available to users since February 2015 [22, 23]. PTS system can cooperate with Phoromatic tool operating in client-server architecture and can be used to remotely manage tests on a group of computers. Tests can be automatically invoked at predefined time schedule, so we can collect data on performance changes in time.

The usefulness of the PTS system extensions has been verified in many practical experiments. It can be used to find most time consuming kernels as well as to evaluate various interactions. Moreover, we can design long term experiments which are needed to identify software aging effects (software rejuvenation), memory leaks, threats in memory paging systems, etc.

4 Experimental Results

Using PTS system with described extensions and modifications we have performed many experiments with various benchmarks. The capabilities of this approach are illustrated for a sample of executed test sets. We show results for 5 representative test sets:

- T1 - Tests targeted at checking the impact of GCC compiler optimization and the number of used threads – C-Ray benchmark with image generation (http://www.sgidepot.co.uk/c-ray.html); Intel Core i5-4670k (4 GHz, 4 GB RAM).
- T2 - Testing average disk read and write speed using Flexible IO tester benchmark (https://github.com/axboe/fio) for 3 disks: D1 – SSD disk Crucial MX100, D2 – West. Digital WD800BEVS, D3 – Samsung HD080HJ; taking into account different block sizes, sequential and random IO profiles.
- T3 - Testing the effectiveness of graphical processor with Unigine Valley benchmark (https://unigine.com/products/benchmarks/valley/); Intel Core i5-4670k (4.1 GHz, 8 GB RAM, NVIDIA GeForce GTX 750).
- T4 - Data compression tests for *gzip* (https://www.gnu.org/software/gzip/),
- *xz* (http://tukaani.org/xz/) and *bzip2* (http://www.bzip.org/) for different data sources and different compression levels.
- T5 - video coding tests performed on 3 platforms (x264 benchmark, http://www.videolan.org/developers/x264.html): P1 – PC with Intel Core i5-4670k (4.1 GHz, 8 GB RAM, NVIDIA GeForce GTX 750 and OpenGL, SSD disk, Arch Linux),

P2 – P1 without OpenGL, P3 – laptop Intel Core 2 Duo L9400 (1.86 GHz, 4 GB RAM, graphical unit, SSD disk, Arch Linux), P4 – PC Intel Pentium E5200 (2 cores, 2.5 GHz, 2 GB RAM, Seagate disks, Debian GNU/Linux Stretch).

Table 1 presents results (execution times) for test set T1 (performed as one experiment-single session of PTS) in function of 8 compiler optimization levels: O0 – no optimization, O1 and O2 the lowest and standard levels, O2 standard with –*march* = *native* profile, O3 and O3n – high level and high level with native profile, Of and Ofn – aggressive optimization and with native profile. Each test has been executed many times (standard deviation of results was in the range 0.05–0.35%). The best results have been obtained for Ofn compilation. The number of threads has a significant impact till 4 threads (system with 8 cores), however the speedup differed to some extent for various optimization levels. Increasing the number to 16 threads has lower impact (except O2).

Table 1. Average execution times for tests T1 (in seconds)

No. of threads	Compiler optimization level							
	O0	O1	O2	O2n	O3	O3n	Of	Ofn
1	184.03	120.76	108.92	90.72	63.25	49.36	61.48	42.54
4	53.14	35.03	31.29	26.44	18.15	14.26	17.64	12.22
16	49.41	32.55	29.09	24.58	16.72	13.09	16.26	11.26

CPU usage during tests T1 was practically stable at 100% level (25% for each core). However, for 4 threads we observed lowering the CPU usage at the end to the level of 25%, this resulted from faster termination of processing on the remaining threads. This effect disappears in the case of more threads than the number of cores. We have also checked the impact of floating point operations. Eliminating the use of FPU unit instructions (appropriate setting of the compiler and using library *libsoft-fp*) we have got over 212 times higher execution time for O3n level with 4 threads. This confirms high impact of floating point operations.

Tables 2 and 3 present results for test set T2 in relevance to sequential and random IO operations (disk reads and writes), respectively. The presented transmission speed (Table 2) and IO operation rate (Table 3) are shown for 3 disks (D1-D3) and different sizes of data blocks. SSD disk D1 outperforms standard HD disks in transmission speed by several times. Comparing IO rates for random operations we see much higher

Table 2. Average disk transmission speed (MB/s) for tests T2 – sequential operations

Block size	Read operation			Write operation		
	D1	D2	D3	D1	D2	D3
4 kB	462.89	61.10	31.90	311.36	53.96	42.63
16 kB	530.95	61.23	58.78	322.12	59.66	57.41
128 kB	540.38	61.26	55.56	323.57	59.45	57.45
2 MB	538.53	60.69	51.04	319.40	56.54	57.38

Table 3. Average IO operation rate (IOPS) for tests T2 – random operations

Block size	Read operation			Write operation		
	D1	D2	D3	D1	D2	D3
4 kB	94795	201	175	74794	191	159
16 kB	26366	191	169	19016	184	152
128 kB	4249	137	123	2508	121	117

values for D1 disk by several hundreds times as compared with classical disks. This results from head positioning (seeking time) involving mechanical movements. This difference is lower for bigger blocks (lower number of needed head movements). Standard deviation of results was in the range of a fraction to a few percent. In the case of random writes it was higher for D1 disk (up to 7%).

In Fig. 1a and b we present the shape of CPU usage plot and average transmission speed plot during the execution of test T2 for D1 disk (block size 16kB, sequential writes). An interesting issue is that CPU usage is low (below 6%) except some spikes up to 25%. These spikes result from operating system activities (OS system was installed on the tested disk). Similarly, interference of these activities is visible in Fig. 1b showing the transmission speed during the whole test. The presented multiple plots refer to subsequent instances of activated tests.

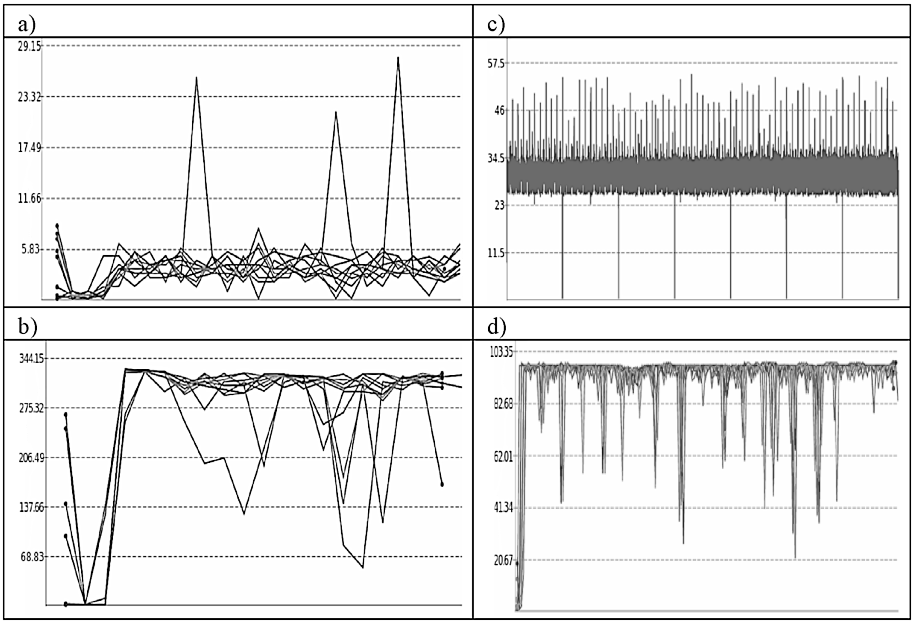


Fig. 1. Performance counter plots for: (a) CPU usage during D1 disk write (blocks 16 kB) in test T2 (y axis 0–29%), (b) disk D1 write transmission speed in test T2 (y-axis 0–344 MB/s), (c) CPU usage for test T3 (y-axis 0–57%), (d) graphical processor usage (y-axis 0–100%) – test T3

In test set T3 we used Unigine-Valley benchmark and verified its operation for 7 different image resolutions: 800 × 600, 1024 × 768, 1280 × 800, 1440 × 900, 1280 × 1024, 1680 × 1050 and 1920 × 1080 pixels. The obtained numbers of generated frames per second were: 87.23, 67,88, 56.75, 48.15, 48.07, 37.98, 33.32, respectively. Hence, the frame generation speed decreases with the number of pixels. However, this follows a non linear relation. Standard deviation of test results was in the range 0.05–0.2%. CPU usage during this benchmark was on average about 30%. It fluctuated (compare Fig. 1c) with spikes ranging to 50%. Graphical processor was used at high level (about 100% with sporadic negative spikes to 20% (Fig. 1d). For increased resolutions the number and amplitude of these spikes decreased.

Table 4 presents results of test T4 related to data compression benchmarks based on *gzip* (uses algorithm DEFLATE), *xz* (uses algorithm LZMA2) and *bzip2* (uses algorithm Burrows-Wheeler) programs for two input data sets (special GNU/Linux files */dev/zero* and */dev/urandom* comprising null characters and pseudorandom data, respectively. Moreover, these tests have been done for different compression levels (1, 6, 9) assuming single tread (because multithreading is supported only for *gzip* and *bzip2*). Standard deviation of results was in the range 0.2–1%. CPU has been used in a stable manner at the average level of 25% (used only one core).

Table 4. Average execution times for test T4 (in seconds)

Block size	Input data sets					
	/dev/zero -1	dev/urandom -1	dev/zero -6	dev/urandom -6	/dev/zero -9	dev/urandom -9
gzip	4.17	26.10	5.48	27.30	5.47	27.76
xz	6.47	216.07	17.30	297.13	17.24	376.91
bzip2	7.56	114.14	7.65	113.09	7.71	115.96

Results of test T5 are summarized in Table 5. The video coder x264 (64 bit edition of x264 in r2666 version) has been used to code (a film available at https://media.xiph.org/video/derf/y4m/in_toTree_444_720p50.y4m) in YUV4MPEG2 format. The effectiveness of coding is measured as the speed of coding in frames per second. Standard deviation of results was in the range 0.2–2% except platform P4 (2.7–7.7%) on which there were some web activities. Processor i5 showed the best results (P1, P2), using OpenCL resulted only in a small improvement (P1 platform). It is worth noting that activation of more threads than CPU cores increases the speed by 15–20%. CPU usage during the test was quite high. For P1 with 4 cores and 4 threads it was in the range 70–84%. Doubling the number of threads increased this value to 100%. Usage of graphical processor fluctuated in the range 6–18%.

Table 5. Average coding speed for x264 (frames per sec) – test T5

Number of threads	System platform			
	P1	P2	P3	P4
1x number of cores	58.72	57.43	9.62	11.40
2x number of cores	70.68	65.98	11.07	13.69
4x number of cores	71.03	66.63	11.00	13.74

Resuming we should notice that the introduced extensions to Phoronix Test Suite gave the capability of performing more complex tests and collecting more information which can facilitate interpretation of system behaviour for different benchmarks.

5 Conclusion

Performance evaluation depends upon selected benchmark programs, test suite management and measurement techniques. Running benchmarks we should collect large amounts of high resolution data about the tested system (or its component) behaviour. Moreover, we should analyse and characterize workload features and their impact on the system performance. Hence, an important issue is not only appropriate system stimuli but also measurements of high level performance parameters (e.g. speed-up, transaction rates) combined with fine grained monitoring (using special counters) integrated with result visualization and reporting. To eliminate the need of benchmark instrumentation for collecting measurement results an efficient test framework is needed. For this purpose we have extended the capabilities of Phoronix Test Suite (87 commits, over 15000 added lines) which is available as open source program [22, 23]. The usefulness of the proposed testing methodology has been illustrated for a representative set of tests.

Further research is targeted at extending the list of performance sensors and extending this capability for other operating system platforms. Moreover, it is reasonable to improve time resolution of the monitoring processes.

References

1. Bienia, Ch., Kumar, S., Singh, J.P., Li, K.: The PARSEC benchmark suite: characterization and architectural implications. Princeton University Technical report TR-811-08 (2008)
2. Chen, T., Guo, Q., Temam, O., Wu, Y., Bao, Y.: Statistical performance comparisons of computers. IEEE Trans. Comput. **64**(5), 1442–1455 (2015)
3. Clemons, J., Zhu, H., Savarese, S., Austin, T.: MEVBench: a mobile computer vision benchmarking suite. In: IEEE International Symposium on Workload Characterization (IISWC), pp. 91–102 (2011)
4. Cooper, B.F., Silberstein, A., Tam, E., Ramakrishnan, R., Sears, R.: Benchmarking cloud serving systems with YCSB. In: ACM Symposium on Cloud Computing, pp. 143–154 (2010)
5. Eigeman, R.: Performance Evaluation and Benchmarking with Real Applications. MIT Press, Cambrige (2001)
6. Feitelstone, D.G.: Workload Modelling for Computer Performance Evaluation. Cambridge University Press, Cambridge (2015)
7. Horn, A.: Kieker: a framework for application performance monitoring and dynamic software analysis. In: ACM International Conference on Performance Evaluation, pp. 247–248 (2012)
8. John, L.K., Eeckhout, L.: Performance Evaluation and Benchmarking. CRC Taylor & Francis, Boca Raton (2006)

9. Kaelli, D., Sachs, K. (eds.): Computer Performance Evaluation and Benchmarking. Proceedings of SPEC Benchmark Workshop. LNCS, vol. 5419. Springer (2009)
10. Kanoun, K., Spainhower, L.: Dependability Benchmarking for Computer Systems. Wiley-IEEE Computer Society Press (2008). ISBN: 978-0-470-23055-8
11. Król, M., Sosnowski, J.: Multidimensional monitoring of computer systems, In: UIC-ATC 2009 Symposium and Workshops on Ubiquitous, Autonomic and Trusted Computing in conjunction with the UIC 2009 and ATC 2009, pp. 68–74. IEEE Computer Society (2009)
12. Kubacki, M., Sosnowski, J.: Applying time series analysis to performance logs. In: Proceedings of SPIE, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments, vol. 9662, pp. 3C1–3C12. SPIE (2015)
13. Kubacki, M., Sosnowski, J.: Performance issues in creating cloud environment. In: Zamojski, W., et al. (eds.) Complex Systems and Dependability. Advances in Intelligent Systems and Computing, vol. 365, pp. 235–244. Springer (2015)
14. Kukunas, J.: Power and Performance: Software Analysis and Optimization. Elsevier, San Francisco (2015). ISBN: 978-0-12-800726-6
15. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: MOTChallenge: towards a benchmark for multi-target tracking (2015). https://arxiv.org/abs/1504.01942
16. Nambiar, R., Poess, M., et al.: TPC benchmark roadmap. In: Selected Topics in Performance Evaluation and Benchmarking, Proceedings of the 4th TPCTC Conference. LNCS, vol. 7755, pp. 1–20. Springer (2012)
17. Nambiar, R., Poess, M. (eds.): Performance Evaluation and Benchmarking, Traditional to Big Data to Internet of Things. Proceedings of the 7th TPCTC Conference. LNCS, vol. 9508. Springer (2016)
18. Ondrejka, R., et al.: Red Hat Enterprise Linux 7 Resource Management Guide. 1.3. Resource Controllers in Linux Kernel. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Resource_Management_Guide/br-Resource_Controllers_in_Linux_Kernel.html
19. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. Neural Netw. **32**, 323–332 (2012). doi:10.1016/j.neunet.2012.02.016
20. Waller, J., Ehmke, N.C., Hasselbring, H.: Including performance benchmarks into continuous integration to enable DevOps. ACM SIGSOFT Softw. Eng. Notes **40**(2), 1–4 (2015)
21. Weiss, C., Westermann, D., Heger, C., Moser, M.: Systematic performance evaluation based on tailored benchmark applications. In: Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering (ICPE 2013), pp. 411–420 (2013)
22. Phoronix Test Suite (2017). http://phoronix-test-suite.com/
23. (2016). https://github.com/phoronix-test-suite/phoronix-test-suite/pull/108 and pages with final parameters /92 and /94
24. Problem Sizes and Parameters in NAS Parallel Benchmarks (2012). https://www.nas.nasa.gov/publications/npb_problem_sizes.html
25. TPC Current Specifications (2017). http://www.tpc.org/tpc_documents_current_versions/current_specifications.asp

Reliability Optimization for Controller Placement in Software-Defined Networks

Jerzy Martyna(✉)

Faculty of Mathematics and Computer Science, Institute of Computer Science,
Jagiellonian University, ul. Prof. S. Lojasiewicza 6, 30-348 Cracow, Poland
martyna@ii.uj.edu.pl

Abstract. Software-Defined Networks (SDNs) are a new networking technology that provides important advantages in IP networking, related to flexibility at network and application levels, especially in control problems. However the SDN prefers a centralized logical control principle. There are still open research issues related to controller placement in SDNs. In this paper a reliability optimization method is presented to solve the problem with multi-controller placements between controllers and network switches. An algorithm was developed that ensures the reliability for deployment of controllers in SDNs.

1 Introduction

Software-Defined Networks (SDNs) enable a clear separation of the data planes from the control [15]. The traditional networks, based on routers, switches, etc. integrate both these planes in the same devices. Contrary to these networks, SDNs propose the application of central controllers that provide central controllers for the development of many control planes to realize a distributed systems [12]. This new network paradigm allows the data forwarding to be simplified, enables full network programmability, and increases the reliability of all network components, etc. Moreover, the SDNs provide designers with a number of challenges, ranging from security, scalability and performance, and ending with improving the reliability of the entire system.

SDN networks have been the subject of much research and numerous studies. Earlier papers presented initial solutions for the SDNs based on the concept of OpenFlow standard [13], later developed in the SDN networks [18]. Further problems associated with SDNs, including controller failures or network disconnections between the control and data planes leading to packet loss and performance degradation have been studied by B. Heller et al. [8]. The adoption of a controller model that acts as a distributed system has been presented by P. Berde [4]. The placement of the controllers that consider some optimization criteria have been analysed in the papers by Bari [2] and Muller [14]. In the first paper dynamic controller provisioning in software defined network was proposed. In turn, in the second paper a placement strategy was introduced which can improve SDN survivability. Nevertheless, none of these proposals considers the controller design up to the optimization of the controller placement in SDNs.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_29

Furthermore, the controller itself might fail and therefore the network nodes might get disconnected from the controller. Moreover, some links can be permanently or temporarily damaged. This will cause the network to lose their efficiency and be deprived of their abilities. It depends on designers to determine what number of defective parts will keep the SDN network at a predetermined high reliability.

Reliability issues of the network have been the subject of numerous studies. Among others, reliability-aware controller placement for SDN networks was presented by Y. Hu [9]. In this paper, through simulation of different placements, algorithms are compared with each other. Network optimization, which aims to achieve high reliability in the network is shown by the same authors in [10]. In this paper a metric was proposed called expected percentage of control path loss to measure reliability. An algorithm to achieve high reliability in the southbound interface between controllers and nodes has been presented by F.J. Ros et al. [16]. Nevertheless, none of the above services apply to multiple controllers, and multiple links in the SDN networks.

The main contribution of this paper is an extension of the optimization problem to multi-controller and multi-link placement in the SDNs. The overall objective is to attempt to solve this problem and provide an algorithm for the sub-optimal optimization with multi-controllers and multi-paths. Moreover, the reliability factors are here defined to define more precisely the reliability of the SDN networks.

The remainder of this paper is organized as follows. Section 2 presents the analytical model of controller placement in the SDN network to optimize the network reliability. The placement heuristic is proposed in Sect. 3. Section 4 provides an approximate reliability evaluation of Software-Defined Networks. The simulation results given in Sect. 5 evaluate the proposed method of solving the problem. Finally, Sect. 6 concludes the work.

2 The Proposed Analytical Model

In this section the reliability optimization framework is described, including problem formulation, reliability lower bound and some results arising from optimization theory.

Figure 1 shows the network architecture of Software-Defined Network with its distributed control plane. The network is classified into three logical plane [17]. The bottom layer, switching layer, is composed of a number of switches which guarantee the distribution of frames among different physical facilities. The middle layer, controller layer, is a centralized entity in translating the requirements from application layer down to the switching layer and propagation the events and local states from switching layer bottom to the application layer. Finally, the top logical layer, application layer, consists here of a number of control applications. In proposed approach all applications can run on multiple physical facilities.

The SDN network is represented by an undirected graph $G(V = N \cup F, E)$, where N is the set of network nodes, F is the set of facility placements where

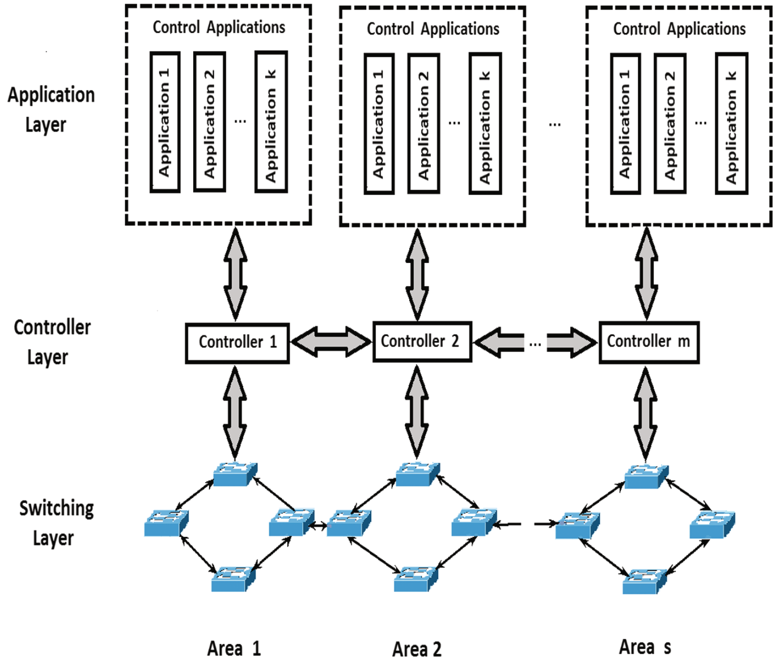


Fig. 1. Architecture of the Software-Defined Network with its distributed control plane.

a controller can be located, E is the set of links among them. Let u, v be the vertices which are connected by means of a directed edge. Thus, $p_{u,v}$ is the probability that the links $(u, v) \in E$ are operational. We assume different i.i.d. operational probabilities for all devices, such as controllers, nodes, and links.

Let ${}_j$ be binary variables that equal 1 if the facility $j \in F$ holds a controller, 0 otherwise. The cost of deploying the controller in such facility is equal to Ψ_j. Let y_{ij} be binary variables that equal 1 if the node i, $i \in N$, connects to the facility j, 0 otherwise. The latency of data transmission to node i connected with the node j is given by L_{ij} which is treated here as a cost of deploying the switch in such facility. Both costs Ψ_j and L_{ij} can be measured in terms more relevant to the network terminology or may also relate to the average economic costs.

The reliability controller placement problem in the SDN can be formulated as follows:

$$\min \sum_{j \in F} \Psi_j x_j + \min \sum_{j \in F} \sum_{i \in N} L_{ij} y_{ij} \tag{1}$$

subject to

$$\sum_{j \in F} y_{ij} > 0 \quad \text{for} \quad \forall i \in N \tag{2}$$

$$y_{ij} \le x_j \quad \text{for} \quad \forall i \in N, \; \forall j \in F \tag{3}$$

$$y_{ij}, \; x_j \in \{0, 1\} \tag{4}$$

$$R(G, i, f_i) \ge \beta \quad \text{for} \quad \forall i \in N \tag{5}$$

where $R(G, i, f_i)$ is the probability defining the reliability between the node i and node f_i, $(f_i \subseteq F)$, β is defined threshold. This means that each node i has to connect to a subset $f_i \subseteq F$ of deployed controller. It is in accordance to the conventional network reliability terminology introduced by M. O. Ball [1] which defines such probability as the k-terminal reliability $R(G, i, f_i)$ between i and f_i, where $k =\mid f_i \mid$.

It is also assumed that is optimized average network reliability, namely

$$\min\{1 - \frac{\sum_{i \in N} \sum_{j \in F} R(G, i, f_i)}{M}\} \tag{6}$$

where M, $M \le\mid N \mid$, is the number of controllers in SDN.

The optimization given in Eqs. (1) and (6) allows to realize the controller placement and optimize the average reliability of the network.

3 Sub-optimal Placement Algorithm to Optimize Network Average Reliability

A challenge when deploying a control infrastructure in the SDN is to determine the control layer, which applies to the designed scenario. Given a network topology, the objective is to find the number of controllers to deploy, and their location. At the same time the reliability of the network must be taken into account, which in accordance with the requirements must get the highest of all possible values.

The proposed optimization follows the backtracking method [5], which is a modification of the search into the greedy approach. The operation of this algorithm is as follows (see Fig. 2). List is also a stack. However, the expansion of the test node is replaced by its extension, here the generation of a single child. If the new node does not meet the criterion of purpose or end, it is further extended with one child. After successive expansions the resulting node meets in the backtracking method the criterion of the end or you cannot allow it to generate new offspring, then it returns to the nearest ancestor to the node, and further expansion is possible.

```
procedure BackTrack ($n$: node; solve: boolean);
begin
  put node $n$ on the stack;
  while the stack is not empty do
  begin
    if the node at the top of the stack is leaf
      if it is a good node and minimize Eq. (1) and Eq. (6)
        return solve $\leftarrow$ true
      else pop it off the stack
      end if
    else
      if the node at the top of the stack has untried children
        push the next untried child onto the stack
      else pop the node off the stack
      end if
    return solve $\leftarrow$ false;
  end;
end;
```

Fig. 2. Pseudo-code of the backtracking procedure used for finding all solutions.

4 An Approximate Reliability Evaluation of Software-Defined Networks

This section presents the results of the approximate reliability evaluation of the SDN networks.

A key issue in the design of interconnection networks, including SDN networks, is fault-tolerance. It is the ability of networks to operate in the presence of components failures [6]. However, the appearance of faults in the network with fault-tolerance should be included in the reliability analysis. It is recalled that the reliability of a SDN network is a function of its topology and the probability of failure of its components and links. To analyse this aspect of reliability, here is introduced a failure model for the SDN.

In practice the parameters of the SDN are normally associated with reliability evaluation and are described by probability distributions. Not all components of the SDN will fail after the same operating time. These times-to-failure obey a probability distribution which describes the probability that a given component fails within a certain specified time or survives beyond a certain specified time.

The controller (switch) reliability can be expressed as the probability of a controller (switch) surviving for a time t, if the i-th controller (switch) failure rate λ_i^c (λ_i^s) is constant, namely

$$R_i^c(t) = exp(-\lambda_i^c t) \tag{7}$$

$$R_i^s(t) = exp(-\lambda_i^s t) \tag{8}$$

Assuming that these components in SDN network can be composed in the parallel systems, gets for different components following dependences:

$$R_{controllers}(t) = 1 - \prod_{i=1}^{M} (1 - exp(-\lambda_i^c t)) \tag{9}$$

$$R_{switches}(t) = 1 - \prod_{i=1}^{S} (1 - exp(-\lambda_i^s t)) \tag{10}$$

where M, S, are the number of controllers and switches, respectively.

The link reliability can be defined as the probability that a link will perform under stated conditions for a stated period of time. Thus, the i-th link reliability $R_i^w(t)$ is expressed as

$$R_i^w(t) = exp(-\lambda_i^w t) \tag{11}$$

where $\lambda_i^w(t)$ is the link failure, i.e. the total number of failures per year for link within year by dividing the number of failures by the lifetime of this link (excluding links in operation for less than 30 days) [19], t is the operational time.

Thus, the reliability of all W links in the SDN network is given by

$$R_{links}(t) = \prod_{i=1}^{W} exp(-\lambda_i^w t) \tag{12}$$

Then, the reliability of all hardware components, including controllers, switches, and links, is given by

$$R_{hardware}(t) = R_{switches}(t) * R_{controllers}(t) * R_{links}(t) \tag{13}$$

In order to find a reliability of the SDN network, the connectivity reliability must be also calculated. Generally, due to a huge computation volume arisen from the exact solution procedures in large networks cannot be used in a realistic scale network. A relevant approximate method is to compute an upper and lower bound of connectivity reliability using minimal cut and path sets [3]. Connectivity reliability of a Origin-Destination (OD) pair [7] indexed by v in the SDN network can be computed as follows

$$R_{connectivity,v} = (R_v^{min} + R_v^{max})/2 \tag{14}$$

where $R_{connectivity,v}$, R_v^{min}, R_v^{max} are respectively average, lower bound, and upper bound of connectivity reliability between OD pair v. To evaluate the connectivity reliability of all OD pairs, the weighted average of OD pair connectivity reliability may be utilized with respect to the ratio of the required demand, namely

$$R_{connectivity} = \sum_{v \in \Omega} \left(\frac{R_{connectivity,v} * q_v}{\sum_{v \in \Omega} q_v} \right) \tag{15}$$

where q_v is the required demand between OD pair v, Ω is a set of all OD pairs. Equation (15) provides a closed formula to evaluate the reliability of network connectivity in the calculation procedure.

Thus, the reliability of a SDN network can be expressed by

$$R_{SDN}(t) = R_{hardware}(t) * R_{connectivity} \tag{16}$$

5 Simulation Results

In this section the simulation results of proposed algorithm for optimization of network reliability are presented.

In the simulation the Internet Topology Zoo [11] was used. or the purposes of the calculation of the simulation experiment was chosen Bren topology, which has 37 nodes. All simulation results were obtained using the Matlab 7.0. All algorithms for the reliability optimization under the assumption of multi-controllers and multi-paths in the SDN networks have been written in this language.

In the first scenario, it is assumed that the simulated SDN network consists of 5 and 10 controllers to be mapped into the topology. The failure rate for all nodes and links is with uniform distribution between 0 and 0.2. The effectiveness of the proposed method was compared to the Simulated Annealing (SA) method. The Simulated Annealing method is used here in optimization problem of placing controllers in a SDN as follows. In each iteration is generated a new candidate solution. It is accepted, if it has a lower cost than the previous. If the new solution is better than the previous, it is associated with an acceptance probability.

Figure 3 shows the average network failure rate versus the failure rate of each node for a defined number of controllers in comparison with the SA method.

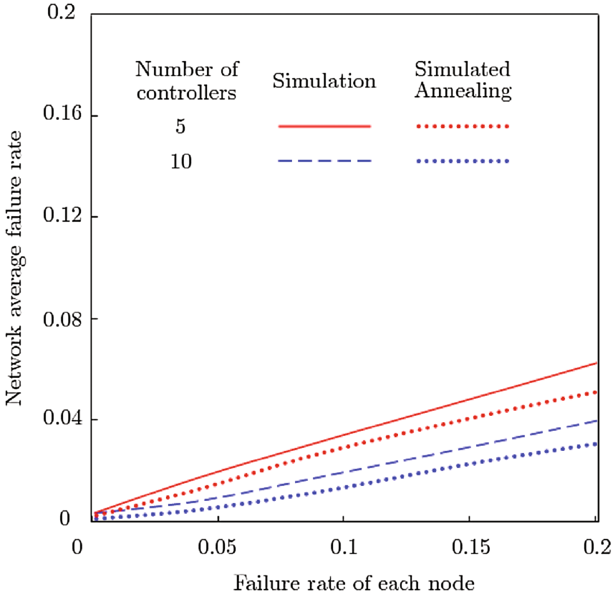


Fig. 3. Average network failure rate versus the failure rate of each node for a defined number of controllers.

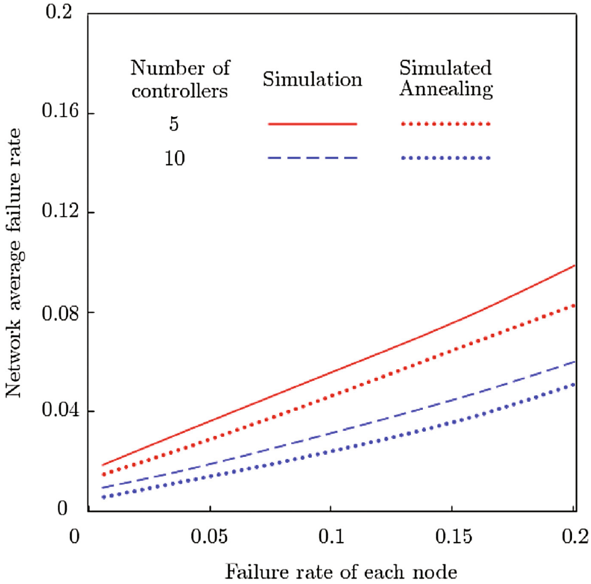


Fig. 4. Average network failure rate versus the failure rate of each node with simultaneous failures of nodes and links for a defined number of controllers.

Reliability was computed according to the criterion given by Eq. (5). Thus the topology is selected with the to generate offspring. This strategy ensures that after the completion of all generated nodes, they are tested. As can be seen from Fig. 3, all of the network average failure rates decrease with the number of controllers for the given algorithm and the SA method.

In the second scenario the failure rate for the SDN network is studied with simultaneous failures of nodes and links with a failure rate with uniform distribution between 0 and 0.2 for both controllers and links respectively (see Fig. 4). As with the previous results, those obtained were compared with the results of the use of SA. The failure rate is slightly higher there than in the previous case. However, the possibility of finding a new link among several does not reduce it significantly. In both cases you can see here a comparison of the effect of the proposed algorithm with the SA method.

In the third scenario, it examined the overall reliability of the SDN network with various number of controllers. Figure 5 illustrates the reliability of the SDN network with 5 and 10 controllers, respectively. With average failure rate of each node equal to 0.08 failures/year, the SDN network with 10 controllers has higher values of the overall reliability over a 6 year period than the network with 5 controllers.

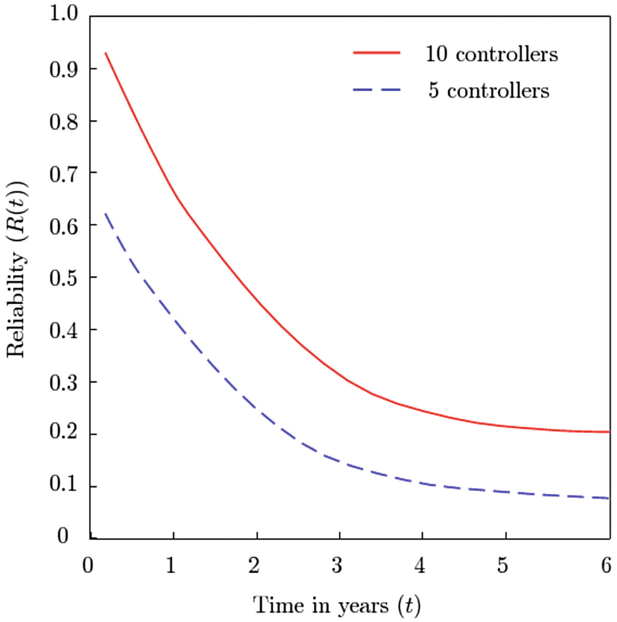


Fig. 5. Graph of analytical performance of the overall reliability of the SDN network with several number of controllers.

6 Conclusion

This paper investigates controller placement in SDN to maximize the reliability of control networks. An information-theoretic model of reliability in the SDN networks has been provided which is useful in network reliability. Unfortunately, the reliability optimization problem in SDN network is an NP-hard problem. In order to simplify the computation complexity, a local optimization algorithm based on the backtracking method has been proposed. Finally, the proposed method of controller placement in the SDN networks has been compared in simulation studies with the simulated annealing (SA) method applied to this problem solution. The simulation results are encouraging: they are close to the results of obtained using SA method.

References

1. Ball, M.O., Colbourn, C.J., Provan, J.S.: Network reliability. In: Technical Research report, TR 92-74, Harvard University (1992)
2. Bari, M., Roy, A., Chowdhury, S., Zhang, Q., Zhani, M., Ahmed, R., Boutaba, R.: Dynamic controller provisioning in software defined networks. In: 2013 9th International Conference on Network and Service Management (CNSM), p. 1825 (2013)

3. Bedford, T., Code, R.: Probabilistic Risk Analysis Foundations and Methodology, 1st edn. Cambridge University Press, Cambridge (2001)
4. Berde, P., Gerola, M., Hart, J., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radosavov, P., Snow, W., Parulkar, G.: ONOS: towards an open, distributed SDN OS. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ser. HotSDN 2014, Chicago, Illinois, USA, pp. 1–6. ACM (2014)
5. Brassard, G., Bratley, P.: Fundamentals of Algorithmics. Prentice-Hall, Upper Saddle River (1995)
6. Duato, J., Yalamanchili, S., Ni, L.M.: Interconnection Networks: An Engineering Approach. Morgan Kaufmann, San Francisco (2003)
7. Guo, D., Zhu, X.: Origin-destination flow data smoothing and mapping. In: IEEE Transactions on Visualization and Computer Graphics (2014). doi:10.1109/TVCG.2014.2346271
8. Heller, B., Sherwood, R., McKeown, N.: The controller placement problem. In: ProcE HotSDN, pp. 7–12 (2012)
9. Hu, Y., Wendong, W., Gong, X., Que, X., Shiduan, C.: Reliability-aware controller placement for software-defined networks. In: Proceedings of the IEEE/IFIP International Symposium on Integrated Network Management, IM 2013, pp. 672–675 (2013)
10. Hu, Y., Wendong, W., Gong, X., Que, X., Shiduan, C.: On reliability-optimized controller placement for software-defined networks. Commun. China **11**(2), 38–54 (2014)
11. Internet Topology Zoo (2017). http://www.topology-zoo.org/
12. Koponen, T., Casado, M., Gude, N., Stribling, J., Pontievski, L., Zhu, M., Roamanathan, R., Iwata, Y., Inoue, H., Hama, T.: Onix: a distributed control platform for large-scale production networks. In: OSDI, vol. 10, pp. 1–6 (2010)
13. McKeown, N., et al.: OpenFlow: enabling innovation in campus networks. ACM Com. Rev. **38**(2), 69–74 (2008)
14. Muller, L.F., Oliveira, R.R., Luizelli, M.C., Gaspary, L.P., Barcellos, M.P.: Survivor: an enhanced controller placement strategy for improving SDN survivability. In: IEEE Global Communications Conference (GLOBECOM), Austin, Texas, USA (2014)
15. Open networking foundation, software-defined networking: the new norm for networks. In: ONF White Paper (2012)
16. Ros, F., Ruiz, P.: Five nines of southbound reliability in software-defined networks. In: Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN, pp. 31–36 (2014)
17. Schmid, S., Suomela, J.: Exploiting locality in distributed SDN control. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networking, ser. HotSDN 2013, Hong Kong, China, pp. 121–126. ACM (2013)
18. Special Report: OpenFlow and SDN - State of the Union (2016). https://www.opennetworking.org/images/stories/downloads/sdn-resources/special-reports/Special-Report-OpenFlow-and-SDN-State-of-the-Union-B.pdf
19. Turner, D., Levchenko, K., Snoeren, A.C., Savage, S.: California fault lines: understanding the causes and impact of network failures. ACM SIGCOMM Comput. Commun. Rev. **40**(4), 315–326 (2010)

Agent Approach to Network Systems Experimental Analysis in Case of Critical Situations

Jacek Mazurkiewicz(✉)

Department of Computer Engineering, Faculty of Electronics,
Wroclaw University of Science and Technology,
ul. Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
Jacek.Mazurkiewicz@pwr.edu.pl

Abstract. Paper presents the analysis and discussion of the network systems in case of the critical situation that happens during ordinary work. The formal model is proposed with the approach to its modeling based on the system behavior observation. The agent approach to constant network monitoring is given using hierarchical structure. The definition of the critical situation sets are created by reliability, functional and human reasons. The proposed method is based on specified description languages that can be seen as a bridge between system description and an analysis tools. Using a multilevel-agent based architecture the realistic data are collected. Analysis is done with a usage of open-source simulation environment that can be easily modified and extended for further work. Based on the simulation results, some alternatives can be chosen in case of system or service failure. This way it is possible to operate with large and complex networks described by various - not only classic – distributions and set of parameters. The presented problem is practically essential for organization of network systems.

Keywords: Network systems · Critical sets · Reliability · Dependability modeling

1 Introduction

Contemporary network systems are very often considered as a set of services realized in well-defined environment created by the necessary hardware and software utensils. The system dependability can be described by such attributes as *availability* (readiness for correct service), *reliability* (continuity of correct service), *safety* (absence of catastrophic consequences on the users and the environment), *security* (availability of the system only for authorized users), *confidentiality* (absence of unauthorized disclosure of information), *integrity* (absence of improper system state alterations) and *maintainability* (ability to undergo repairs and modifications) [1, 2, 12, 19].

The system realizes some tasks and it is assumed that the main system goal, taken into consideration during design and operation, is to fulfill the user requirements. The system functionalities (services) and the technical resources are engaged for task realization. Each task needs a fixed list of services which are processed based on the

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_30

system technological infrastructure. The different services may be realized using the same technical resources and the same services may be realized involving different sets of the technical resources. It is easy to understand that the different values of performance and reliability parameters should be taken into account. The last statement is essential when tasks are realized in the real system surrounded by unfriendly environment that may be a source of threads and even intentional attacks.

The presented work uses the agents in task of the transportation system monitoring and modeling, so we propose the following description of the most important agent's features [4]:

- unique identification within the proposed architecture,
- interaction abilities and proper interfaces for communication and different data transfer,
- secure protocols necessary for communication purposes,
- hardware and/or software implementation,
- plug-and-play ability to guarantee promising scalable and flexible structure.

The temporary computer engineering still does not define an "agent" term in detailed way, but it is not a real barrier to establish the unified semantic meaning of the word in technical point of view. The agent can play the role of the autonomous entity [5] as a model or software component for example. The agent's behavior can be noticed as trivial reactions, but is not limited – so we can easily find agents characterized by complex adaptive intelligence. Sometimes is important to point the potential adaptive abilities of the agents [6]. It means the agent can gather the knowledge from the environment around and to tune their behavior as a reaction for different events.

This way we can say the agents belong to the softcomputing world. The agent's structure is not obligatory plain. We can easily [4] find at least two levels (lower, higher) of the rules created for the agents. This approach allows to tune the level of the sensitivity for the environment and to define the vitality feature of the agent understood as activity or passivity [10, 14].

We propose to use the agents to create the intelligent hierarchical monitoring architecture - described in Sect. 4. The Sect. 5 presents a solution of a description language for a proposed model, called *SML* (*System Modeling Language*). As a format of the proposed language *XML* (*Extensible Markup Language*) was chosen. Main reason is a simple (easy to learn) and readable structure, that can be easily convert to text or other format. Moreover, *XML* is supported not only with various tools (providing validation possibilities) but is also supported by many programming languages and framework in case of quicker and more efficient implementation.

The aim of this paper is to point the problems of the critical situations in unified network system – product of essential elements and features taken from the real systems: *Discrete Transport System (DTS)*. Each part of the system is characterized by unique set of features and can caused the critical situation of whole system if it starts to work in unusual way or the fault or error of it is noticed. It is hard for an administrator, manager or an owner to understand the system behavior and to combine the large scale of variant states of it in single – easily observable and controlled global metric as a pointer to make the proper decision in short time period. To overcome this problem we propose a functional approach. The system is analyzed from the functional point of view, focusing on business service realized by a system [20]. The analysis is following a classical [14]: modeling and simulation approach. It allows calculating different system measures, which could be a base for decisions related to administration of the transportation systems. The results of the system observation – understand as the set of data collected during the simulation process are the basis to define the critical situations and they allow providing proper solution to lift-up the systems in effective way if the critical situation occurs. This is the only sensible way, because the critical situations are the real and not removable part of the system life. The organization of this paper is as follow. We start with description of the abstract service network model and its simulation rules (Sect. 2). Base in it we define the quality performance metric (Sect. 3). In Sect. 6 we provide the most adequate – in case of the level of detail - the well-established description of the critical situation. Section 7 presents the case study and the related results.

2 Network Model

A realization of the transportation system service needs a defined set of technical resources. Moreover, the operating of vehicles transporting commodities between system nodes is done according to some rules – some management system. Therefore, we can model discrete transportation system as a 4-tuple [15]:

$$DTS = \langle Client, Driver, TI, MS \rangle \tag{1}$$

where: *Client* – client model, *Driver* – driver model, *TI* – technical infrastructure, *MS* – management system.

Technical infrastructure includes set of nodes with defined distances between them and a set of vehicles. Each vehicle is described by its load (number of containers) and random parameters which model vehicle breakdowns (requiring repair by one of the maintenance teams) and traffic congestion (which result in random delays in the transportation time). The service realized by the clients of the transport system is mail sending from some source node to some destination one. Client model consists of a set of clients. Each client is allocated in the one of nodes creating the transportation system [18]. The client allocated in an ordinary node generates containers (a given amount of commodities, measured in discrete numbers) according to the Poisson process with destination address set to ordinary nodes. In the central node, there is a set of clients, one for each ordinary node. Each client generates containers by a separate Poisson process and is described by intensity of container generation.

The human infrastructure is composed by the set of drivers. So the description of this part of system infrastructure requires the analysis of the drivers' state and the algorithms, which model the rules of their work. Each driver could be in one of following states (s_d): rest (not at work), unavailable (illness, vacation, etc.), available (at work – ready to start driving), break (during driving), driving. The number of driver working hours is limited by the labour law. The daily limit for each driver equals to 8 h and a single driver operates with one truck. Drivers work in two shifts, morning or afternoon one.

Moreover we propose to categorise the driver's illnesses as follows: short sick: 1 to 3 days, typical illness: 7 to 10 days, long-term illness: 10 to 300 days [11]. We prepare the daily record of the driver. The decisions (send a truck to a given destination node) are taken in moments when a container arrives to the central node. The truck is send to a trip if: the number of containers waiting in for delivery in the central node of the same destination address as that just arrived is larger than a given number, there is at least one available vehicle, the simulated time is between 6 am and 22 pm minus the average time of going to and returning from the destination node.

The truck is send to a node defined by destination address of just arrived container. If there is more than one vehicle available in the central node, the vehicle with size that a fits the best to the number of available containers is selected, i.e. the largest vehicle that could be fully loaded. If there are several trucks with the same capacity available the selection is done randomly. The restriction for the time of truck scheduling (the last point in the above algorithm) are set to model the fact that drivers are working on two 8 h shifts.

2.1 Network Systems Simulation Tools

There are various methods of system analysis. Some researches try to do it using graphs [8], others choose some simulation techniques. In this paper we consider system behavior that will be as close to reality as is can be. Usage of simulator allow us to mimic the behavior of a system. In literature, two main types of simulators can be found: a continuous time and discrete event based simulation [10, 17]. Continuous simulation requires a representation of the system using differential equations [8]. This type of simulator is predominately related with electric power studies. For this reason it will be excluded from further research. The other group of simulators are discrete events that describe the system behavior as a series of events. Classically discrete event simulators are basis for telecommunication and IT analysis tools.

The set of the most popular simulators of this kind is as follows: *OPNET* [4, 16] and *NS-2* [13], both well known by stakeholders, as well as *QualNet* [1], *OMNeT++* [6], *SSFNet/PRIME SSF* [20], and *SGOOSE* [8]. Experiments reported in this paper were performed using the *SSFNet* simulation environment developed by the Renesys Corporation with support from *DARPA*. *SSFNet* has large number of protocols models and network elements; moreover open-source code allows modification. In this paper Java based version of *SSFNet* was used since Java language allowed much faster development then a usage of C++. *SSFNet* simulator consists of three major parts: *SSF* engine, *DML* language [7] and *SSFNet* models.

The *SSF* (*Scalable Simulation Framework*) is public-domain standard for discrete-event simulation implemented in C++ with Java and C++ interface. Scalable Simulation Framework is a base for higher level - the *SSFNet*. *SSFNet* module is a collection of Java packages for modeling and simulation of networks and Internet protocols. Moreover *SSFNet* uses public-domain standard called *DML* (Domain Modelling Language) to configure simulation scenarios. For the purpose of this work some extensions were developed, mainly connected with support for traffic generation (models of user behavior), simulation of business level services, implementation of resource consumption for requests processing. Since fault and failures models are integral part of dependability analysis the *SSFNet* was extended to in-corporate errors. Errors were introduced in different levels beginning from link failures, network adapter failures to software component failures [20].

Additional modules of the tool required the extension of its input language (*DML*) used in standard *SSFNet* version, but the most important extension was implementing Monte-Carlo approach [20] based on running simulation several times and calculating results based on averages values. In this way - during each simulation - the parameters described in by stochastic processes - were the traffic generation which modeled user behavior in a random way. They have different values (according to set-up distributions) including an influence on the system behavior. The capability of multiple runs of simulation was added to standard *SSFNet* package by changes in several *SSFNet* classes (setting up random seed and clearing all static collections). Results of simulation are recorded in specified output file that allows further post-processing in case of dependability metrics.

3 Quality Performance Metric

The quality of the system is assessed by its ability to transport commodities on time. Due to assumed grain of model system we observe the containers (a given amount of commodities) during simulation. The container is assumed to be transported on time when a time measured from the moment when the container was introduced to the system to the moment when the container was transferred to the destination is smaller than a guaranteed time of delivery.

To measure the quality of the whole system we propose to use ratio of on-time deliveries in 24 h time slots. Let $N_d(t)$ denotes the number of containers delivered in the period the day t, and $N_{pd}(t)$ denotes the number of delivered containers on time within the same 24 h. Therefore, the system quality could be measured by a ratio of on-time deliveries, defined as:

$$a_t = \frac{N_{pd}(t)}{N_d(t) + 1} \tag{2}$$

The denominator includes +1 modification to prevent the ratio go to infinity in case of a full stoppage of the system (i.e. no containers delivered in the analyzed period).

4 Monitoring Architecture

In case of Monitoring Architecture representation and distributed multilevel agent-based architecture can be constructed. Figure 1 shows the diversification of complexity of a system into layers and their placement in a system. The lowest components of the structure are *Node Probes* (*NP*) which are the simplest pieces of the architecture representing resident level. These are the simplest and easy to get data that at this level represent small value that is why they are aggregated in upper units are

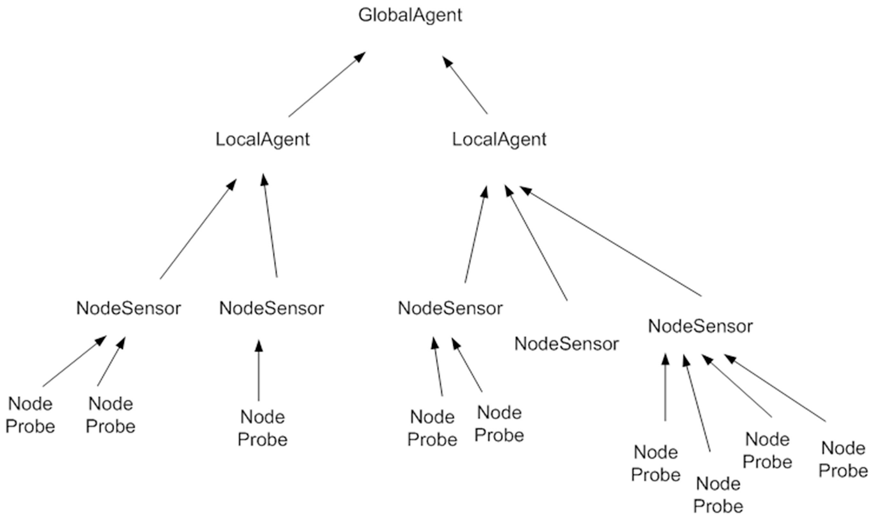


Fig. 1. Multilevel architecture schema

forwarded to appropriate supervising *Node Sensors* (*NS*). Next *Node Sensors* collect the data and create an image of the particular area – so they are located in the ordinary nodes (*ON*). Again the information is sent to a higher level – *Local Agent* (*LA*) – combined to the central node (*CN*).

$$NS_i = \bigcup_j NP_j; j \in N \tag{3}$$

This set of information creates a database building representation of local part of a system (subnetwork). It means that the local view of the system and partial administration in the system can be done at this level.

$$LA_i = \bigcup_j NS_j; j \in N \tag{4}$$

The highest component of this structure is the *Global Agent* – working in the headquarter (*HQ*), that picks and process local information's and view to one central unit.

$$GA = \bigcup_j LA_j; j \in N \tag{5}$$

This module stores all information from a whole system. It is situated in one point and one dedicated machine (with a strong backup).

Assembling all local view at this level we get one homogenous global view. At this level, data-mining techniques can be used. We can see that set of information flow goes to the central unit – *Global Agent*. For this reasons it is the most complex and the simplicity of the data that are needed to describe the system in this point is the highest in hierarchy.

5 Description Language

Since, the purpose of the work is to analyze network system based on specified mathematical model, there is a need to transfer the data into a format that would be useful in an analysis tool.

It requires specify data format that can be easily shared between various tools or even several of transport architectures (independent form complexity). Several data sharing and exchange standards have been developed in the Intelligent Transport Systems [9].

They define a standard data format for the sharing and exchange of the transportation data mostly based on *UML* (*Unified Modeling Language*) diagrams. Other solutions, i.e. Japanese standard called *UTMS* (*Universal Traffic Management Systems*) focuses rather on the road traffic system.

Still none of them is coherent with solutions proposed in this paper, since they describe different types of network system. Moreover they are based on *UML* diagrams, which are the graphical representation of a model, but not the one, that can be simply used as an input format for any available analysis tool (computer simulator). Additionally description language for this system should be as close as real, not only to a mathematical description of the system, but to real system behavior and its parameters. In Sect. 4 we mentioned that the view of the network system can be realized on two levels (local and global). To do that, the tool for visualization and data processing is needed. Furthermore having this tool we can not only see the topology of the system, but also its elements and parameters. It gives us an opportunity to see the system more precisely or even make some analysis on a real data that comes from proposed multilevel agent-based architecture. Still, it requires specify data format that can be shared between tools, but since of the data exchange is done based on *UML* diagrams, there is a need used some other solution that will be more suitable. Since *UML* diagrams are mostly graphical representation of a model, we propose a solution of a *description language* for a proposed model, called *SML* (*System Modeling Language*) [15]. Format of this language is based on *XML* standards, since it is easy to use, and extendable. Moreover the format allows using the language without special tools, since *XML* is supported by many tools. Figure 2 shows a fragment of the language with appropriate elements and attributes related with mathematical model described previously.

As can be seen, each element of the system is modeled as a complex element with appropriate sub-elements and attributes. The proposed language assures aggregation of dependability and functionality aspects of the examined systems. One language describes whole system and provides a universal solution for various techniques of computer analysis as an effective and suitable input for those tools. Expect easiness of potential softcomputing analysis, promising scalability and portability (between analysis tools) can be named as the main advantage of the language usage. Proposed language is easy to read and to process using popular and open-source tools; however the metadata used in this format are still a significant problem in case of file processing (large size of the file).

```
<Node>
         <SingleCentralNode to="Wroclaw">
          <numberOfPackages>3455</numberOfPackages>
          <numberOfVehicles>8990</numberOfVehicles>
          <ManagementSystem />
          <TechnicalInfastuctureTopology numberofOrdinaryNodes="5819">
                   <timeBetweenSpecificNodes>
                       <linksBetweenNodes from="Wroclaw" to="Opole" />
                        <time>5.7</time>
                    </timeBetweenSpecificNodes>
                </TechnicalInfastuctureTopology>
            </SingleCentralNode>
        </Node>
        <Vehicle>
            <meanspeed>9.7</meanspeed>
            <capacity>678</capacity>
            <MTTF>10.05</MTTF>
            <MRT>50.98</MRT>
        </Vehicle>
```

Fig. 2. *SML* – fragment of the language for *DTS*

Nevertheless since *XML* format is strongly supported by programming languages like: *Java*, *C#*, the usage as much as processing of the file can be done irrespectively from application language. As previously described (Fig. 1) data send by *Node Probe* are combined in the *Node Sensors*. Each of these entities has assigned to it a supervisor – *Local Agent* that accumulates these files in order to create local view. This level is more compound and computational complex than previous one considering installed database and some methods that solve additional problems. In this way *XML* files transferred from the simplest level to the next one – creating views on the upper level. *Global Agent* collects this information and similarly to Local Agent combines all information included in dedicated *SML* file. As this Global Agent is the most resourceful entity it may be distributed, so it can contain more than one database. At the end, full description of the system is created, visualized and analyzed with respect to dedicated analysis tools.

6 Critical States of Operation

The word "critical" is linked to the term of "crisis" which refers to a "change of state", "a turning point" [3] or "being at a turning point, or a sudden change" [8]. It seems the universal definition of the term is not easy and maybe is not possible. The most proper and as close as possible approach to the description of the system situation is based on the systems attributes and weighted combination of reliability and functional features of it. For the discrete transport systems (*DTS*) discussed in the paper the three quality states of the system are defined: operational, failed or critical. This is our new approach to the problem of the critical state description – completely different to our previous works [11].

The aim of this paper is to show a method that will allow to predict the system behavior within a short time (few days) horizon knowing the actual condition described by the results of multi-criteria analysis focused on the owner point of view. It means the goal is to foresee the most sensible direction of the system management or maintenance decisions or operations.

6.1 System Functional and Quality States

The functional state of the system S_t at the end of each day t is given by a 3-dimensional vector that includes: the number of drivers nd_t, that are not sick, the number of vehicles nv_t that are operational and number of stored containers in the warehouses nc_t.

$$S_t = [nv_t, nd_t, nc_t]. \tag{6}$$

We propose to analyze the system its functional state to one of three quality states: operational, critical and failed. It is done based on assessing the performance metric defined in (2). Moreover, we planned to use this assessment to predict the system behavior within a short time horizon Δt (few days). Therefore, we propose to assess a functional state as operational one when the probability that a system will fulfill the performance requirements (i.e. the acceptance ratio will be larger than required level α on $t + \Delta t$ *day*) is larger or equal to a given threshold level θ.

In a similar way, the system state is assumed to be failed if the probability that a system will not fulfill the performance requirements within a given time horizon is larger or equal to a threshold θ. All other states are assumed to be critical one. Let's introduce a more formal definition. For two thresholds α, $\theta \in (0.5, 1)$, a given functional state S_t of system at day t is named as:

$$\begin{array}{cl} \text{operational} & \text{if } P(a_{t+\Delta t} > \alpha) \geq \theta \\ \text{failed} & \text{if } P(a_{t+\Delta t} \leq \alpha) \geq \theta \\ \text{critical} & \quad\text{otherwise} \end{array} . \tag{7}$$

In other words, after noticing that $P(\alpha_{t+\Delta t} \leq \alpha) = 1 - P(\alpha_{t+\Delta t} > \alpha)$, we can define the critical state as a state for which:

$$1 - \theta < P(a_{t+\Delta t} > \alpha) < \theta. \tag{8}$$

7 Test Case Analysis

We propose for the case study analysis an exemplar *DTS* based on the Polish Post regional centre in Wroclaw. We have modeled a system consisting of one central node (Wroclaw regional centre) and twenty two other nodes - cities where there are local post distribution points in Dolny Slask Province.

The length of roads were set according to real road distances between cities used in the analyzed case study. The intensity of generation of containers for all destinations were set to 4.16 per hour in each direction giving in average 4400 containers to be transported each day. The vehicles speed was modeled by Gaussian distribution with

50 km/h of mean value and 5 km/h of standard deviation. The average loading time was equal to 5 min. There were two types of vehicles: with capacity of 10 and 15 containers. The MTF of each vehicle was set to 20000. The average repair time was set to 5 h (Gaussian distribution). We also have tried to model the drivers availability parameters. We have fulfilled this challenge by using the following probability of a given type of sickness - short sick: 0.003, typical illness: 0.001, long-term illness: 0.00025. The tests were realized for the acceptance ratio level $\alpha = 0.95$, $\Delta t = 2$ days and threshold level $\theta = 0.8$. Moreover the number of drivers that are not sick are taken from the set $nd_t \in (85, 120)$, the number of vehicles that are operational is spanned $nv_t \in (45, 60)$ and number of stored containers in the warehouses $nc_t \in (4000, 7000)$ (Fig. 3).

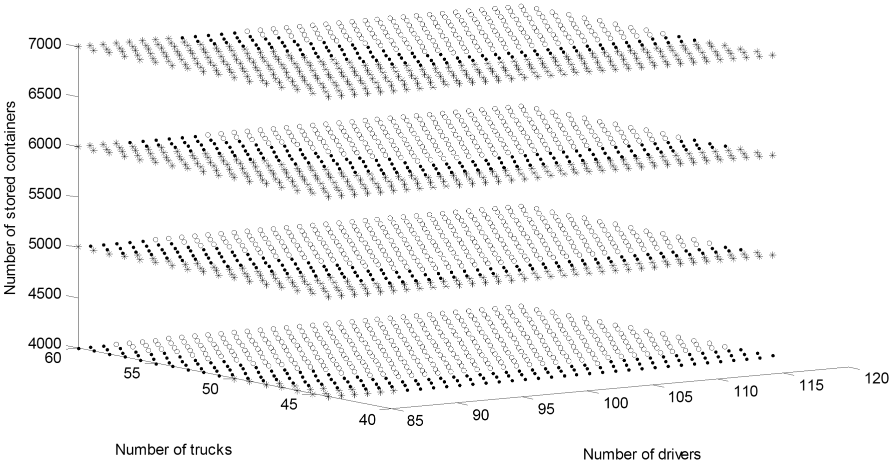


Fig. 3. Assignment of states (triple: number of trucks, drivers and stored containers) to operational (circles), critical (dots) and failed (asterisks) group

7.1 Results

The results presented in Fig. 3 could be used as an indicator for management decision. If at the end of given day the system is in a state (defined by a number of operational trucks, working drivers and number containers stored in central and ordinary points) that is assigned in Fig. 3 to a critical group (marked as dot) it should raise an alarm for the management.

Occurrence of such state indicates that the probability that within a few days the performance quality will drops below thresholds raises. The system manager could react by increasing the system resources (drivers and/or trucks). The quantitative analysis of such reaction was presented by authors in [11].

8 Conclusions

We have presented a formal model of sophisticated network system including reliability, functional parameters as well as the human factor component at the necessary level of detail. The model is based on the essential elements and features extracted from the *Discrete Transport System (DTS)*. We pointed the crucial conditions of the normal work of the system. The critical situation is described and discussed to create the Pareto set – guarantying the possible safety operating points for actual network system.

The proposed approach allows performing reliability and functional analysis of the different types of network systems – for example:

- determine what will cause a "local" change in the system,
- make experiments in case of increasing volume of the commodity incoming to system,
- identify weak point of the system by comparing few its configuration,
- better understand how the system behaves.

Based on the results of simulation it is possible to create different metrics to analyze the system in case of reliability, functional and economic case. The metric could be analyzed as a function of different essential functional and reliability parameters of network services system. Also the system could be analyze in case of some critical situation (like for example a few day tie-up [21]).

The presented approach – based on two streams of data: dependability factors and the features defined by the type of business service realized – makes a starting point for practical tool for defining an organization of network systems maintenance. It is possible to operate with large and complex networks described by various – not only classic – distributions and set of parameters. The model can be used as a source to create different measures – also for the economic quality of the network systems. The presented problem is practically essential for defining and organization of network services exploitation.

References

1. Al-Kuwaiti, M., Kyriakopoulos, N., Hussein, S.: A comparative analysis of network dependability fault-tolerance, reliability, security, and survivability. IEEE Commun. Surv. Tutorials **11**(2), 106–124 (2009)
2. Avizienis, A., Laprie, J.C., Randell, B.: Fundamental concepts of dependability. LAAS-CNRS Research Report No. 1145, LAAS-CNRS, Toulouse, France (2001)
3. Barlow, R., Proschan, F.: Mathematical Theory of Reliability. Society for Industrial and Applied Mathematics, Philadelphia (1996)
4. Bonabeau, E.: Agent-based modelling: methods and techniques for simulating human systems. In: Proceedings of National Academy of Sciences (2002)
5. Gao, Y., Freeh, V.W., Madey, G.R.: Conceptual framework for agent-based modelling and simulation. In: Proceedings of NAACSOS Conference, Pittsburgh (2003)
6. Jennings, N.R.: On agent-based software engineering. Artif. Intell. **117**(2), 277–296 (2000)

7. Kołowrocki, K.: Reliability of Large Systems. Elsevier, Amsterdam-Boston-Heidelberg-London-New York-Oxford-Paris-San Diego-San Francisco-Singapore–Sydney-Tokyo (2004)
8. Kyriakopoulos, N., Wilikens, M.: Dependability and Complexity: Exploring Ideas for Studying Open Systems, EN. EC Joint Research Centre, Belgium (2001)
9. Liu, H., Chu, L., Recker, W.: Performance evaluation of ITS strategies using microscopic simulation. In: Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems, pp. 255–270 (2004)
10. Mascal, C.M., North, M.J.: Tutorial on agent-based modelling and simulation. In: Winter Simulation Conference (2005)
11. Mazurkiewicz, J., Walkowiak, T.: Analysis of critical situation sets in discrete transport systems. In: Kabashkin, I.V., Yatskiv, I.V. (eds.) 12th International Conference: Reliability and Statistics in Transportation and Communication RelStat 2012, Riga, Latvia, 17–20 October 2012, pp. 354–361. Transport and Telecommunication Institute (2012)
12. Mazurkiewicz, J., Walkowiak, T., Nowak, K.: Fuzzy availability analysis of web systems by Monte-Carlo simulation. In: LNCS, LNAI, pp. 616–624. Springer, Heidelberg (2012)
13. Melhart, B., White, S.: Issues in defining, analyzing, refining, and specifying system dependability requirements. In: Proceedings of the 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2000), 3–7 April 2000, pp. 334–340. IEEE Computer Society, Edinburgh (2000)
14. Mellouli, S., Moulin, B., Mineau, G.W.: Laying down the foundations of an agent modelling methodology for fault-tolerant multi-agent systems. In: ESAW 2003, pp. 275–293 (2003)
15. Michalska, K., Mazurkiewicz, J.: Functional and dependability approach to transport services using modelling language. In: Jędrzejowicz, P. et al. (eds.) LNCS/LNAI, LNAI, vol. 6923, pp. 180–190. Springer, Heidelberg (2011)
16. Michalska, K., Walkowiak, T.: Modeling and simulation for dependability analysis of information systems. In: Świątek, J. et al. (eds.), Information Systems Architecture and Technology. Model Based Decisions, pp. 115–125. University of Technology, Wroclaw (2008)
17. Walkowiak, T., Mazurkiewicz, J.: Algorithmic approach to vehicle dispatching in discrete transportation systems. In: Sugier, J., et al. (eds.) Technical Approach to Dependability, pp. 173–188. Wroclaw University of Technology, Wroclaw (2010)
18. Walkowiak, T., Mazurkiewicz, J.: Analysis of critical situations in discrete transport systems. In: Proceedings of International Conference on Dependability of Computer Systems, Brunow, Poland, 30 June–2 July 2009, pp. 364–371. IEEE Computer Society Press, Los Alamitos (2009)
19. Walkowiak, T., Mazurkiewicz, J.: Availability of discrete transportation system simulated by SSF tool. In: Proceedings of International Conference on Dependability of Computer Systems, Szklarska Poreba, Poland, June 2008, pp. 430–437. IEEE Computer Society Press, Los Alamitos (2008)
20. Walkowiak, T., Mazurkiewicz, J.: Functional availability analysis of discrete transportation system simulated by SSF tool. Int. J. Crit. Comput. Based Syst. **1**(1–3), 255–266 (2010)
21. Walkowiak, T., Mazurkiewicz, J.: Soft computing approach to discrete transportation system management. In: LNCS, LNAI, vol. 6114, pp. 675–682. Springer, Heidelberg (2010)

Reliability Assessment of Driving Systems of City Buses

Marek Młyńczak[1](✉), Murat Muzdybayev[2], Alfiya Muzdybayeva[2], and Dinara Myrzabekova[2]

[1] Faculty of Mechanical Engineering,
Wrocław University of Science and Technology, Wrocław, Poland
marek.mlynczak@pwr.edu.pl
[2] East Kazakhstan State Technical University, Ust Kamenogorsk, Kazakhstan
muratmuzdybaev@gmail.com,
{amuzdybaeva,dmyrzabekova}@ektu.kz

Abstract. Paper concerns reliability assessment of very sensitive subsystems from the point of view of safety of city buses analyses maladjustment of buses quality to difficult Asian operating conditions. There are discussed various factors describing operational conditions of city buses and motivation of reliability assessment of selected bus subsystems. Data base, covering times to failure of buses, was created upon real observations and enables for experimental verification of complex approach taking into account tribo-technical and statistical analysis. Failure analysis has shown weak elements in chassis and suspension. Probability failure distribution, reliability, density and hazard rate functions are assessed for leaf springs and wheel studs. Statistics is performed for mileage to first and second failure. Analysis of failure causes of some elements is shown.

Keywords: Reliability assessment · Bus · Driving system

1 Introduction

Public transport is an important infrastructure in modern agglomerations. The main objectives of existing that system is: making easy mobility for citizens and mitigation of congestions created by private transport. Public transport effectiveness and safety depends on many factors covering population, culture, traffic density, traffic infrastructure condition, geo and climatic conditions, technical state of means of transport. Quality of transport system depends on many factors making first of all fast and comfortable movement according to stated schedules. Main factors influencing transportation system and process is geo-localization and population distribution over the agglomeration area. Those factors state main assumptions to transportation organization. Secondary factor is a decision of transportation operator related to design a transportation system and choosing a means of transport. Usually transportation systems are based on various vehicles like: buses, trams, metro, cable line, magnetic trains, moving platforms, etc. Decision depends on costs covering initial costs like: design, infrastructure construction, vehicle purchase, setting up organization, staff training, etc.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_31

and operational costs dealing with current maintaining of the system (fuel/energy, salaries, taxes, insurance, service, repairs, materials, modernizations etc.) [4, 6].

The paper describes influence of bad vehicle quality on decreasing of transportation effectiveness. An example of transportation system mentioned in the paper shows necessity of vehicles improvement due to low reliability resulting in low availability, cancelling courses and lowering effectiveness of the transport measured in passenger-kilometers [6]. Bad reliability means here high failure frequency, low repairability and low travelling comfort resulting from pure design and law-cost constructional materials [2].

2 Research Background

Many external factors influence the performance of automotive materiel but it is obvious that one of the most important is condition of road surface (type of cover, roughness, texture, cracks, ruts, etc.). City buses operated in inadequate road conditions will result fast in failures of suspension assemblies and chassis which are the most attackable by working load and road surface [4]. Discovering the most frequent failures of construction elements will allow for analysis of failure causes and offer scientifically grounded solutions to improve vehicles reliability in the further process of operation. According to this statement, it is assumed that critical parts of buses would be elements of suspension and chassis. After investigating their failure-free time of operation it will be possible to discover the least reliable structural elements [7]. That knowledge may help in rational ways of their durability extending and, consequently, improving reliability of units themselves and suspension assemblies and chassis in a whole [1–3, 5].

Research objective is to develop common research method:

- to investigate failure – free operation time of suspension and chassis of city buses,
- to perform failure analysis of buses and discover the most frequent failure units of suspension and chassis,
- to analyze failure modes and their causes to offer some design and material based methods making them more resistant to operational and environment impacts,
- to improve reliability of frequently failed structural elements of suspension and chassis to extend their durability.

Proposed approach is based on data collected from transportation operator running a fleet of “average” quality buses. Buses are operated in the uniform environmental conditions specific to one of the Asian middle-size cities located in difficult geographical and weather conditions (rough road surface, slopes, high range and gradient of temperature and humidity, dusty and polluted air). The park of city buses in Ust – Kamenogorsk became morally and physically off the map. Operator, “Ulba-Transport” LLP were the pioneers who step by step made attempts to solving this problem. They chose city public transport bus of Yutong vehicle brand, in particular 6108 HGH model

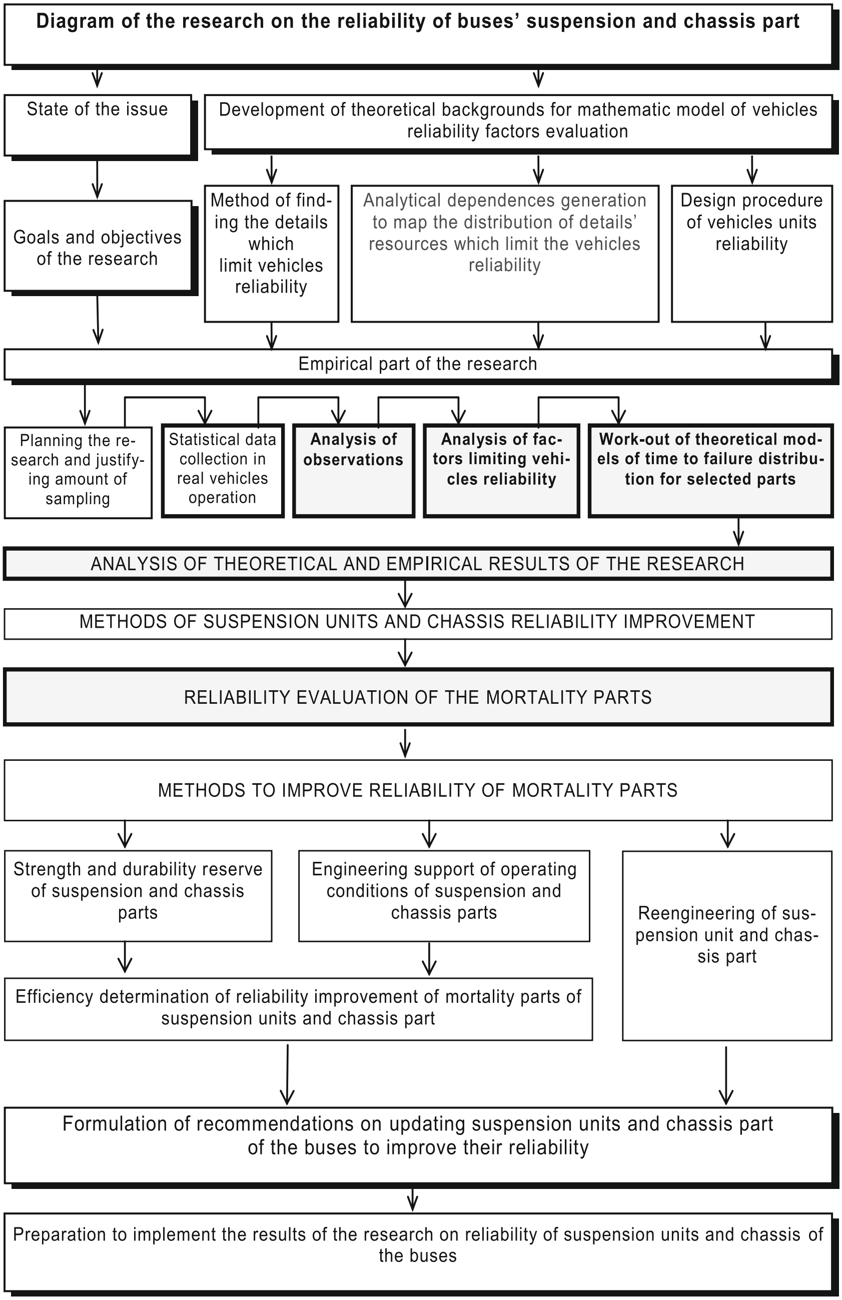


Diagram of the research on the reliability of buses' suspension and chassis part
State of the issue
Development of theoretical backgrounds for mathematic model of vehicles reliability factors evaluation
Goals and objectives of the research
Method of finding the details which limit vehicles reliability
Analytical dependences generation to map the distribution of details' resources which limit the vehicles reliability
Design procedure of vehicles units reliability
Empirical part of the research
Planning the research and justifying amount of sampling
Statistical data collection in real vehicles operation
Analysis of observations
Analysis of factors limiting vehicles reliability
Work-out of theoretical models of time to failure distribution for selected parts
ANALYSIS OF THEORETICAL AND EMPIRICAL RESULTS OF THE RESEARCH
METHODS OF SUSPENSION UNITS AND CHASSIS RELIABILITY IMPROVEMENT
RELIABILITY EVALUATION OF THE MORTALITY PARTS
METHODS TO IMPROVE RELIABILITY OF MORTALITY PARTS
Strength and durability reserve of suspension and chassis parts
Engineering support of operating conditions of suspension and chassis parts
Reengineering of suspension unit and chassis part
Efficiency determination of reliability improvement of mortality parts of suspension units and chassis part
Formulation of recommendations on updating suspension units and chassis part of the buses to improve their reliability
Preparation to implement the results of the research on reliability of suspension units and chassis of the buses

Fig. 1. Diagram of the research (shaded blocks deals with that paper)

(China made). This bus is manufactured under the license of MAN and perfectly serves the principle of “quality- price” from the perspective of climate conditions in Ust – Kamenogorsk. This fact worked as the choice foundation.

Operational problems have appeared not long after buses introduction to operation especially with the suspension and chassis what directed an attention to the problem of application wrong construction material to these units.

Preliminary works on reliability were directed on testing material used to produce bus suspension units [7]. It was analyzed problems of automotive materiel reliability in general as well as the influence of external and internal effects on reliability and durability of transport equipment [1]. Another issue dealt with an influence of operating (daily distances, load, speed, etc.) and maintenance (quality of preventive and correction maintenance) aspects on availability. Finally, it was done tribo – technical tests of friction elements came from breaking system [7]. The above approach in the form of algorithm is shown in Fig. 1. It is used methodology based on reliability improvement described in [3, 8]. Operational data obtained from operator is confident unfortunately on demand of that operator.

3 Statistical Analysis of Selected Bus Units

Raw analysis of buses suspension and driving unit failure was based on operational data and shows that more than 90% of failures are caused by wheel stud and leaf springs (Fig. 2). These elements have been selected to further statistical analysis.

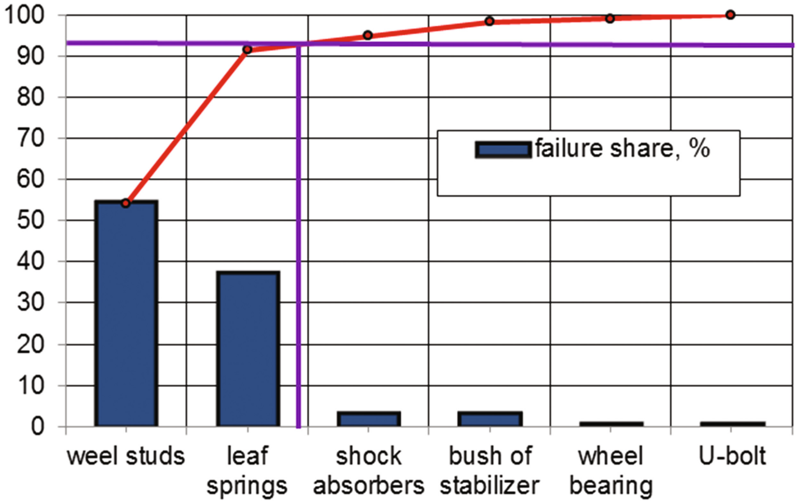


Fig. 2. Pareto distribution of suspension units and chassis of tested buses

Field test data i.e. mileage to failure in [km] was processed to determine failure distribution function. Sample size was taken assuming accuracy of sample characteristics is ε = 0,10 ÷ 0,15 and belief probability α = 0,90 ÷ 0,95 [8].

It was assumed: normal, lognormal, Erlang, Weibull and exponential distributions. The best approximation was obtained by normal (time to first failure) and Weibull distributions (time to second failure) respectively. Results of approximation of theoretical distributions of mileage to failure for above selected parts of suspension and chassis are given in Figs. 3, 4 and 5. Figure 3 presents failure distribution, reliability functions, density function and hazard rate function for time (mileage) to first failure for front (a) and rear (b) leaf spring.

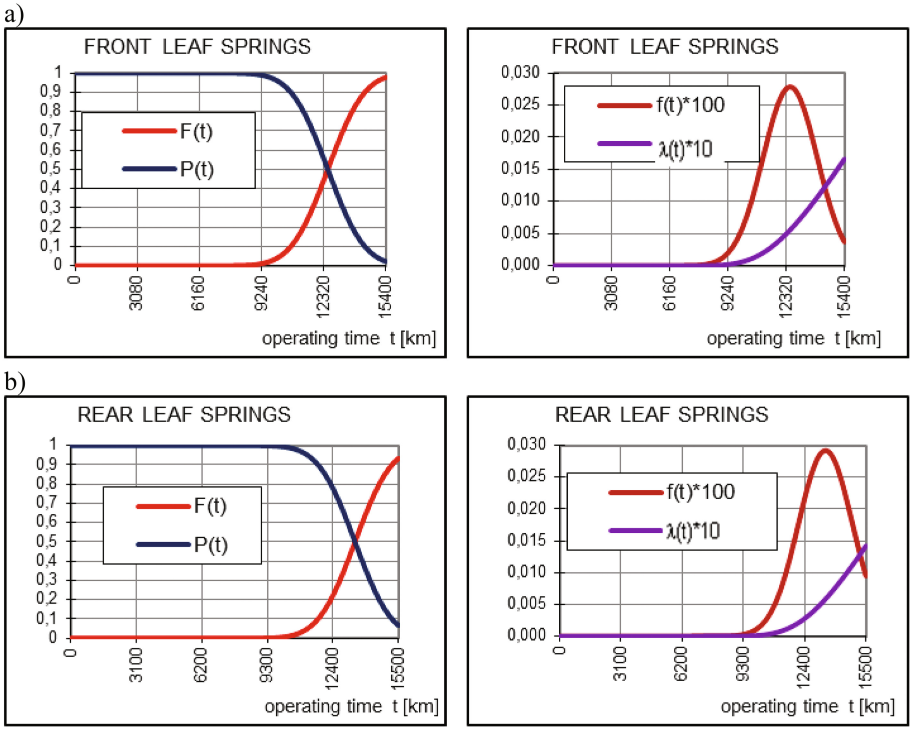


Fig. 3. Theoretic lifetime distribution for front (a), and rear (b) spring to the first system failure P(t) – reliability function, F(t) – probability of failure, f(t) – probability density function, λ(t) – failure rate

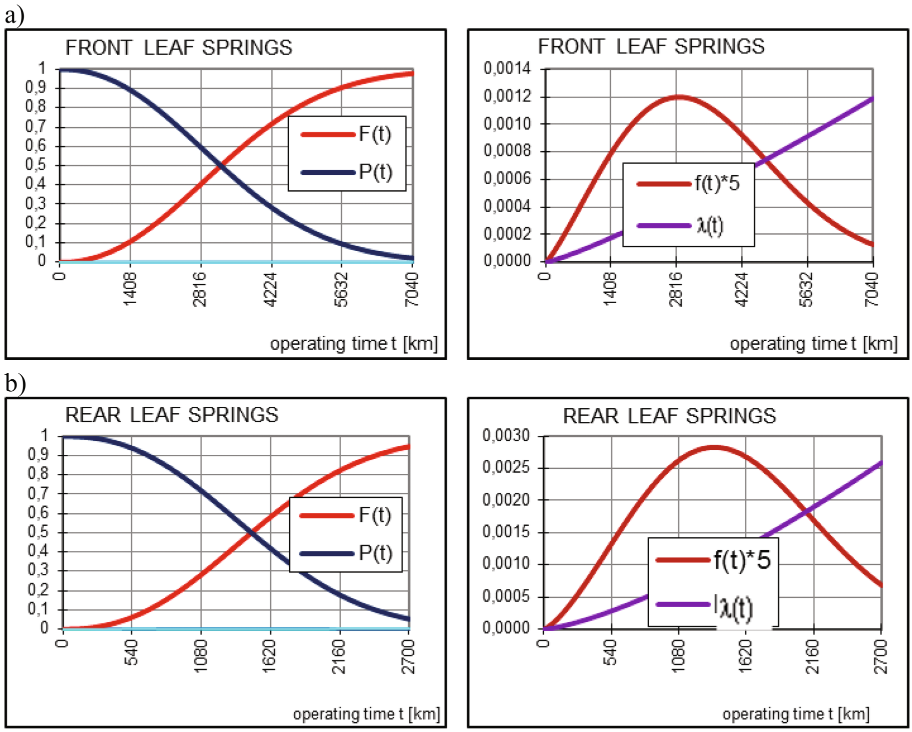


Fig. 4. Theoretic lifetime distribution for front (a), and rear (b) spring to the second system failure. P(t) – reliability function, F(t) – probability of failure, f(t) – probability density function, λ(t) – failure rate

Figure 4 shows similar relations for time to second failure for front (a) and rear (b) leaf spring. Figure 5 presents failure distribution and reliability functions for time (mileage) to first and second failure for wheel stud.

Table 1 presents statistical details of obtained theoretical models of reliability assessment of analyzed elements of suspension and chassis.

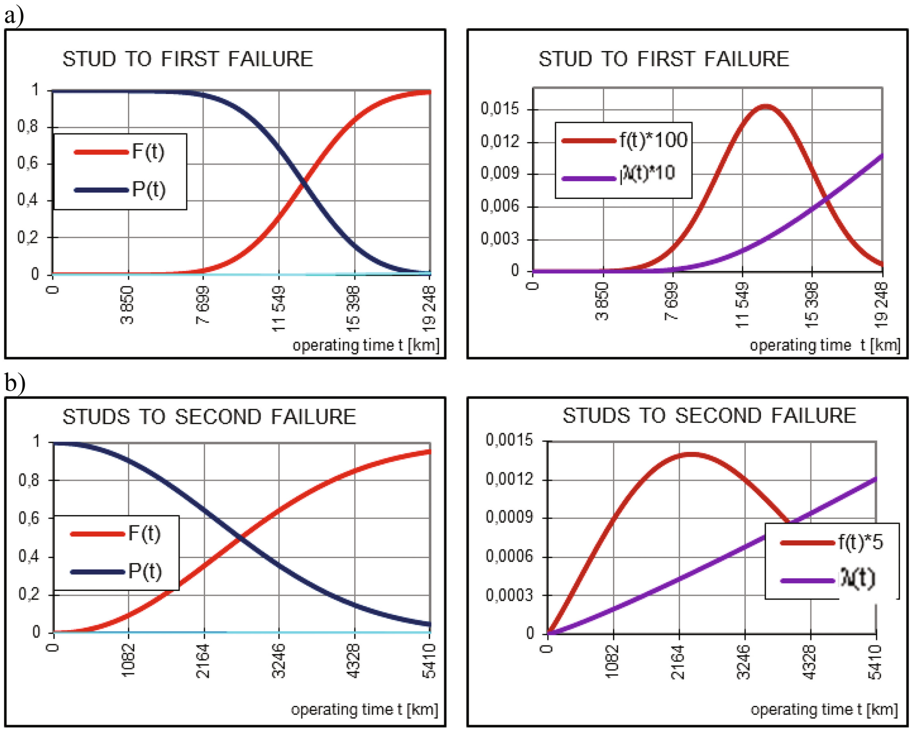


Fig. 5. Theoretic lifetime distribution of wheel stud to first (a) and second failure (b) P(t) – reliability function, F(t) – probability of failure, f(t) – probability density function, λ(t) – failure rate

Table 1. Statistical parameters of analyzed distributions

Bus element name	Time to first failure		Time to second failure	
	MTTFF [km]	Distribution & Param.	MTTSF [km]	Distribution & Param.
Front leaf springs	12404	N (12529;1432)	3204	Weibull (2,18;3797)
Rear leaf springs	13545	N (13453;1365)	1490	Weibull (2,38;1717)
Wheel stud	12801	N (12803;2597)	2676	Weibull (2,13;840)

MTTF – mean time to first failure; MTTSF - mean time to second failure

4 Failure Analysis

It is specified that failure of leaf spring is caused by fatigue cracks (Fig. 6) and subsequent breaking in the crack area in a spring (Fig. 7).

It’s been established that basic factor causing these failures is accumulated damage and fatigue of material [4], as well as excessive load occurring in difficult road conditions, i.e. due to improper condition of the road (in some places there is no roadbed at all). Improper condition of roads causes relatively high rate of impact loads and Yutong 6108 bus suspension was not designed for such a load. Significant factor of failure occurrence is uncontrolled tightening of spring by U-bolt during maintenance work. The latter can cause additional stress in this element of suspension where central bolt is located.

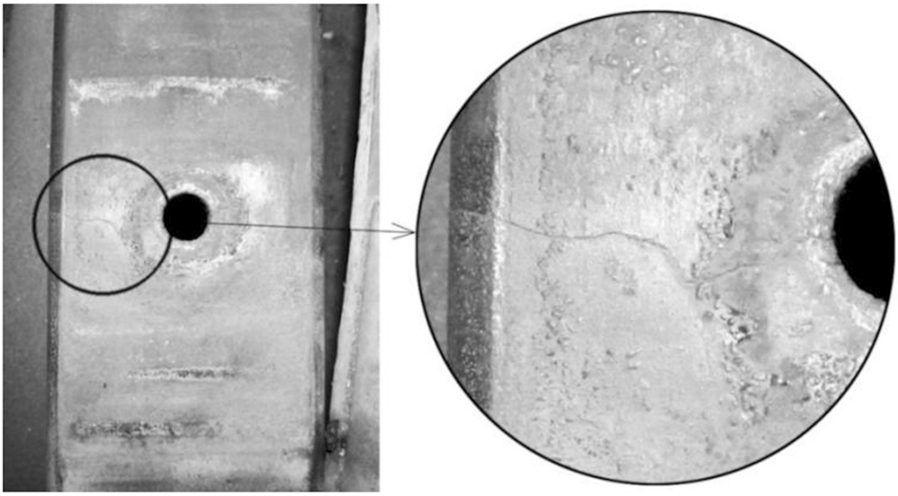


Fig. 6. Crack in leaf spring

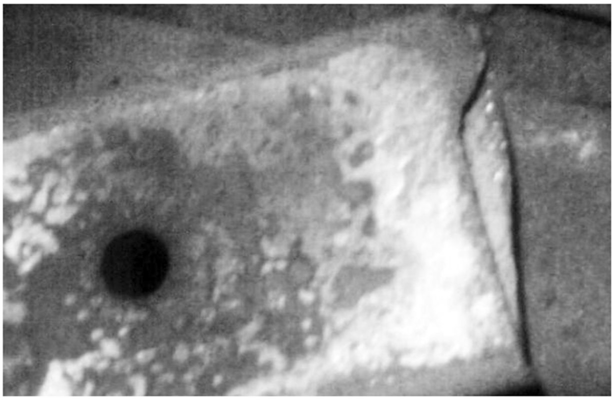


Fig. 7. Part of leaf spring after breaking

Another weak element causing the most significant bus failures (Fig. 2) is wheel stud (Fig. 8). Frequent breaking of that element results in bus stopping and withdrawing it from the usage.

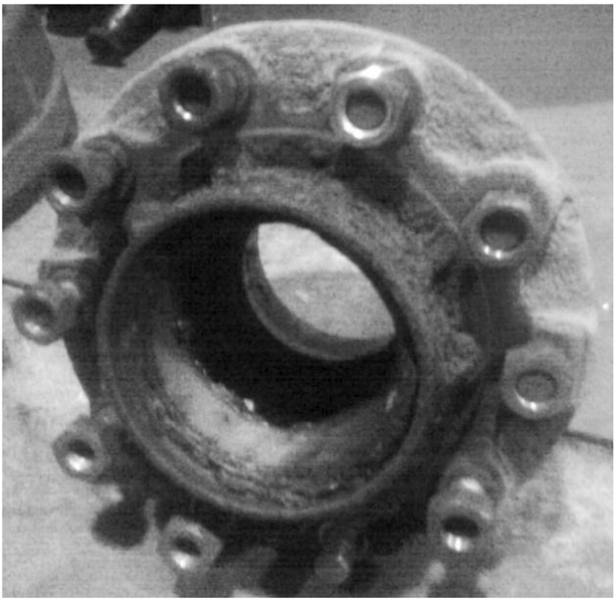
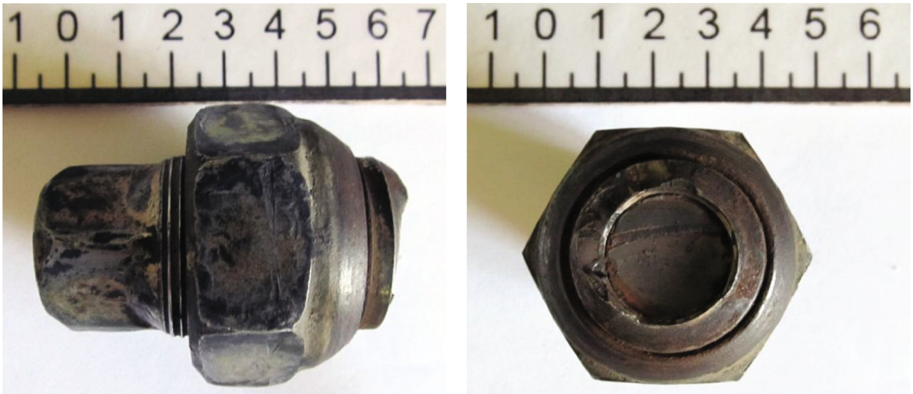


Fig. 8. Wheel part of Yutong 6108 bus with studs

It is observed that stud failures are due to cracks and subsequent breaking in threaded part of the stud, exactly on the edge between the joint and threaded sleeve that tighten stud (Fig. 9).



side view

axial view

Fig. 9. Stud with inner cap nuts and outer cap nuts after breaking

Analysis of stud breaks (Fig. 9, axial view) indicates brittle fracture of continuous structures with high hardness [4]. That hypothesis as the most credible one was taken as the basis for further research and element improvement idea.

Hardness tests of stud material shows that studs have relatively high hardness in breaking zone both on the surface and in the center (Fig. 10).

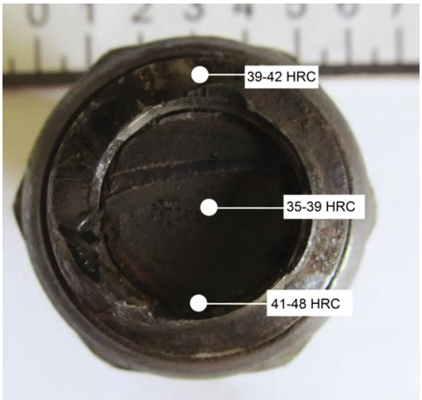


Fig. 10. The results of hardness measurement

Hardness measured on the surface of threaded segment in the center of cross section shows 35-39 HRC. Thus, results of measurements prove suspicion that studs are prone to brittle fracture. It has been also identified that there is no maintenance documentation for technical service of wheel mechanisms. So the excess of studs tightening torque in wheel replacement can cause additional stress in the stud and it's breaking during dynamic loads in the process of movement along road irregularities.

5 Conclusion

It is presented problem of low reliability and durability of buses operated in difficult environmental conditions. Information obtained from the transportation company has initiated theoretical and experimental research aiming to rising busses reliability and transportation system performance. Research methodology has been proposed which covers: field test observations, data collecting and processing, failure analysis, material and tribo – technical tests and statistical analysis of operational data to evaluate reliability characteristics. Failure modes of the least reliable elements of suspension and chassis part were analyzed and the most probable causes of their failures were identified. Statistical models of failure – free mileage of critical components are presented.

References

1. Barringer, H.P., Weber, D.P., Life cycle cost tutorial. In: Fifth International Conference on Process Plant Reliability. Gulf Publishing Company, Huston (1997)
2. Bentley, J.P.: Introduction to Reliability and Quality Engineering. Addison-Wesley Longman Ltd., Harlow (1999)
3. Blischke, W., Murthy, D.N.P.: Reliability: Modeling, Prediction, and Optimization. John Wiley & Sons, Inc., New York (2000)
4. Gołąbek, A. (ed.): Busses Reliability. Wydawnictwo Politechniki Wrocł, Wrocław (1993
5. Haugen, E.B.: Probabilistic Mechanical Design. Willey, New York (1980)
6. Kutz, M. (ed.): Handbook of Transportation Engineering. McGraw-Hill Companies, Inc. (2004)
7. Muzdybaev, M.S., Muzdybaeva, A.S., Rogovskij, V.V., Yrzaвeкova, D.M.M, Elemes, A.B., Wieleba, W., Lesniewski, T.: Preliminary experimental research of tribological properties of bus brakes friction linings. Tribologia (2016)
8. Szor, J.B., Kuzmin, F.I.: Reliability Assessment of Technical Appliances. WNT, Warszawa (1970)

Testing the Significance of Parameters of Models Estimating Execution Time of Parallel Program Loops According to the Open MPI Standard

Łukasz Nozdrzykowski and Magdalena Nozdrzykowska(✉)

Maritime University of Szczecin,
ul. Wały Chrobrego 1-2, 70-500 Szczecin, Poland
{l.nozdrzykowski,m.wrobel}@am.szczecin.pl

Abstract. The authors present their own models for estimating the execution time of program loops in parallel systems with message transmission (Open MPI) along with an analysis of the significance of the parameters used for the estimation and for proving the correctness of these models, which is the main goal of this article. The significance analysis proposed by the authors of the models was based on measurements of actual performance of test loops from the NAS Parallel Benchmarks, which provides sets of test loops in the field of numerical problem solving. By applying the significance analysis, we demonstrate the accuracy of selected parameters as relevant parameters of the presented models. The significance analysis was based on soft reduction of conditional attributes using the relative probability of rules useful in the theory of rough sets.

Keywords: Program loops · Time estimation · Significance analysis · Rough set theory

1 Parallel Programming with Message Transmission

Today, with multi-core processors easily available, we can take advantage of parallel programming to have operations performed in parallel, which by assumption should accelerate calculations. However, when parallelization at the level of multi-core processor gives unsatisfactory acceleration or the parallel program runs too long, we can utilize the processing power of other computers connected together by a fast network. To implement the parallel program on multiple networked computers, we can use the common MPI standard used when writing applications for these systems [1]. Open MPI (Message Passing Interface) is an open implementation of the protocol used for passing messages between the processes of parallel programs performing on one or more computers. In this standard messages are sent between the hosts involved in the calculations, transmitting calculation and other data and the results from the individual hosts. [2] The Open MPI is used for parallel programming and the programming of distributed systems. According to [3], both systems are separated by the addressing system. In the case of distributed systems there is no common address space, therefore

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_32

in multi-core machines with local memory and computer networks [4]. In this paper, the authors use programming in Open MPI for parallel systems.

Implementation of the parallel program using several computers can speed up the performance of a parallel program. However, it is also important to note that the transmission of large amounts of data may also slow down the execution time of the entire application, especially in the case of Open MPI, where messages are sent. In this case, it may lengthen the entire process so that parallelization in accordance with the Open MPI will be unprofitable. Therefore, it is necessary to estimate the execution time of the operation in the parallel system according to the chosen standard of programming and assess whether further parallelization in the established manner makes sense. In the article, the authors focus on estimating the execution time of the program loop in Open MPI, because the loop is the longest part of the program performance and their parallelism gives the greatest benefits.

The use of the Open MPI to get the program loops in parallel requires the transformation of those program loops in which there are data dependencies so that these dependencies are removed [5]. For this purpose, we can use loop transformations to the sequential-parallel form [6], which honor data dependencies allowing parallelization of the loop part that includes no data dependencies. Data dependencies occur due to asynchronous operation of threads in the parallel section, which may change the order of operations and thus produce incorrect calculation results by the parallel program. In the case of program loops, the following types of write-read dependencies may occur (Fig. 1a), where data are written to variable X in the source code, and in the subsequent lines of the source code variable X is read. Another data dependency is write-write (Fig. 1b), where in various lines of the source code various data are written to variable X. The third type of read-write dependency (Fig. 1c) is where data are read from the variable X, then in subsequent lines of the source code a different value is written to the variable X [7].

a. X=a Y=X	b. X=a X=b	c. Y=X X=a

Fig. 1. Types of data dependencies occurring in the source code

Program loops can be divided into three types compatible with specific FAN, PAR and PIPE transformations, proposed and named by Lewis T. in [8]. In loops compatible with the FAN transformation there are no dependencies between instructions within the loop body. A FAN implements a structured fork/join control flow operation and refers to SIMD calculations. The parent process P1 sequentially creates processes P2, P3,…, Pk, where each process executes its identical code on its own data (different for each process) and returns a value to the process P1. [8] Another loop type is one compatible with the PAR transformation, where dependencies between iterations occur, and they may not occur inside the loop body. The PAR performs one or more independent line of code in parallel loop body, where each process works on identical data, then it synchronizes the parallel code execution through a barrier at the end of each iteration, and repeats this pattern [8]. The third type of loop is compatible with the PIPE transformation, where mixed dependencies occur inside the loop body and between iterations. In the PIPE transformations k processes are needed to execute k pipe statements in parallel, each process copies all k statements, and executes them sequentially. For loops, consistent with the PIPE transformation we propose to divide the loop into parts compatible with FAN or PAR transformation [8].

2 Estimating the Loop Execution Time in Parallel Systems with Message Transmission

The decision on the use of parallel programming using Open MPI and assigning tasks to multiple computers should be made only if the task execution time so done can be satisfactorily shortened. The methods coming to assistance in such cases are those able to estimate the particular loop execution time without having to physically run them. Well-defined models for time estimating should be capable of doing so with acceptable precision.

To estimate the runtime of program loops in parallel in multi-computer systems, in [9] the author proposed a model for the OpenMP standard and adapted it to the OpenMPI standard. The proposed model estimates the execution time of the program loop consistent with the FAN and PAR transformations in the standardized Open MPI and presents the accuracy of the estimates. The complexity of the model (multiple parameters) requires that the correctness of these models should be verified. To prove the correctness of these models, we determined the significance of the parameters of these models, making use of the rough set theory. The first model is prepared for program loops compatible with the FAN transformation. This model is described by Eq. (1).

$$T(n) = \max_{2 \le j \le J} \left(\sum_{k=1}^{K} \frac{r_{jk} \cdot li_j \cdot z_k}{n_j \cdot lp_j} + w_j \cdot md_j + cw_j \right) + ti \tag{1}$$

The model has a maximum function indicating the longest run-time of calculations from individual hosts participating in the calculations marked *J*. The computation time for individual hosts is the sum of the individual data localities (the number of *K* depends on the tested loop) from the product r_{jk} meaning the execution time of a simple operation with *k-th* data locality multiplied by the number of iterations *li* and the number of operations with *k-th* data locality marked z_k. The product is divided by the number of threads *n* and the number of pipeline stages of *j-th* processor lp_j. Added to that sum is $w_{j,}$ the product of the time of data transmission to *j-th* host and the number of data necessary to calculate the md_j and the time of synchronization of threads on the j-th host cw_j. Finally, the time of measurement initiation *ti* is added, depending on the chosen method of measuring the loop execution.

The second model is prepared for program loops compatible with the FAN transformation. For PAR transformation, the model of time estimation compatible with Open MPI is described by Eq. (2).

$$T(n) = \max_{\substack{1 \le j \le J \\ 1 \le h \le H}} \left(\sum\nolimits_{k=1}^{K} \frac{r_{jhk} \cdot li_h \cdot z_{hk}}{n_h \cdot lp_h} + w_{jh} \cdot md_{jh} + cw_{jh} \right) + ti \tag{2}$$

In the PAR transformation the loop is divided into independent fragments that are performed by different threads. The threads execute different portions of the source code on the same data set. For the PAR-compatible model we introduced an additional maximum function of variable h denoting an independent fragment of the loop processed by a thread, where the program loop is divided into H fragments. Other parameters are the same as in the Eq. (1).

Models (1, 2) contain two types of parameters: those dependent on the testing environment and those characterizing the tested program loop. Parameters characterizing the test environment are as follows: r - execution time of a simple operation on the processor, w - time to send a single character between threads, n - number of threads, lp - number of pipeline stages on the processor, cw - time of thread synchronization, ti - measurement initiation time.

The parameters characterizing the tested loop are these: li - number of iterations, z - number of operations, k - data locality, md - number of data needed for calculations for one thread. The method of determining parameters characterizing the environment was presented for Open MP and Open MPI standards in the article [9]. Due to the high variability of resources available in current computer systems we should measure the environmental parameter performance each time. This will allow for time estimation accuracy dependent on the current load of computing power. The parameters characterizing the program loop should be defined by the programmer alone or with the appropriate tools automating this step, for instance those based on the polyhedral representation. Likewise, the same code of program loop should be previously parallelized as per the Open MPI. However, whether it is worth using the possible parallelization with the available resources or not is decided on the basis of above proposed models.

We compared the proposed models of time estimation for FAN and PAR transformations in Open MPI standard to actual measurements of loop execution taken from NAS benchmarks [10]. In the article [9] the authors showed that the estimation error made by the proposed models was up to 16.47%. The mean estimation error by the models was 8%. In the next section, the authors intend to prove the correctness of their models by analyzing the significance of model parameters.

3 Significance Analysis of Time Estimation Models Parameters

A large number of parameters in the proposed models makes it necessary to verify relevant parameters and reject those of low significance. We can make use of the NAS benchmark [10], which provides sets of test loops in the field of numerical problem solving. The provided program loops have irregular memory access, often including multi-dimensional array variables. In addition, it features enhanced communication.

Using program loops from the NAS benchmark we ran tests of these loops and obtained loop execution time and all parameters used in models of time estimation in parallel systems.

The assessment of parameter significance in reference to the presented models was based on soft reduction of conditional attributes using the relative probability of rules useful in the theory of rough sets [11, 12]. Through the use of soft reduction of conditional attributes we gain an opportunity for rejecting attributes whose elimination does not reduce the number of useful rules [13]. This allows conducting quality assessment of individual rules. We use here the relative probability of atomic rules expressed by the formula (3).

$$P_w = \frac{P}{L} \tag{3}$$

Where

P - sum of the probabilities of useful atomic rules,
L - number of elementary conditional sets.

Each atomic rule is considered to be useful when its probability is greater than a predetermined threshold. The significance analysis is based on the determination of decision and conditional attributes, together with their appropriate coding. Then individual conditional attributes are reduced and their significance is determined on the basis of the number of generated certain rules. Soft reduction of conditional attributes leads to a slight decrease in the number of entirely certain rules.

The analysis of the significance of the attributes of the authors models for loop execution time estimation are compatible with FAN and PAR transformations for parallel systems with message transmission based on 22018 program loops from the NAS benchmark.

The test platform consisted of three computers with Intel Core i7 processors providing a total of 24 hardware threads. These computers were connected by the fast Gigabit Ethernet. The calculations were performed on two computers: MSI GE72-2QD Intel Core i7-5700HQ clocked at 2.7 to 3.5 GHz and on one computer with Intel Core i7-2600 clocked at 3.4 GHz. All computers used 16 GB of DDR3 memory, and SSD Kingston SM2280S3120G drives on the M.2. port. The network connection was implemented through the managed TP-Link TL-SG105E switch.

The rules were analyzed using the relative probability of 80 and 90%. Two values were taken for observing changes in levels of significance. In this study we decided to verify the significance for higher probability that leads to more rejections of uncertain rules. In such case it is more difficult to obtain a higher level of significance, which proves the correct choice of attributes (model components).

The program loop execution time was chosen as the decision attribute, while specific parameters of time estimation models were conditional attributes. Because the Open MPI standard and the network connection are used, we can assume that the highest level of significance will characterize the attributes associated with communication, above or equal to the significance level of attributes characterizing the loops themselves and computer parameters. For the analysis each attribute had to be coded.

This encoding was performed using the method of equal number of samples in the intervals. Each attribute was encoded into five intervals. It is assumed that the more intervals, the better adjustment to reality is. However, the number of intervals should be chosen to match the number of rules, indicating the influence of conditional attributes on the decision attribute. The division into intervals should be such that each rule occurs more than once. For the presented test environment, the coding was performed in accordance with Table 1.

Table 1. Attribute coding.

Attribute	Intervals accepted for coding
Execution time	<0; 0.0099), <0.0099; 0.1182), <0.1182; 0.152), <0.152; 0.25), <0.25; ∞)
r	<0; 0.00000000049); <0.00000000049; 0.0000000027), <0.0000000027; 0.00000024), <0.00000024; 0.00000084), <0.00000084; ∞)
w	<0; 0.00176), <0.00176; 0.07), <0.07; 0.24), <0.24; 0.5), <0.5; ∞)
lp	<1; 5>, <6; 9>, <10; 15>, <20; 24>
li	<1; 30000), <30000; 145000), <145000; 600000), <600000; 1400000), <1400000; ∞)
cw	<0; 0.00081), <0.00081; 0.052), <0.052; 0.111), <0.111; 0.221), <0.221; ∞)
md	<1; 2>, <3>, <4; 5>, <6; 7>, <8>
z	<1; 7>, <8; 10>, <11; 15>, <16; 20>, <21; ∞)

With so adopted coding of conditional attributes and the decision attribute, the significance analysis covered all of the conditional attributes. Because attribute *tw* is practically constant, its significance was not analyzed. At a constant value, its significance is relatively low, and the attribute can be reduced. The analysis was based on a comparison of the relative probabilities for all useful atomic rules to the relative probability of all useful atomic rules reduced for each conditional attribute. In order to estimate the former, we determined the number of elementary sets of useful atomic rules and the relative probability of all useful atomic rules for a non-reduced set of conditional attributes. The results are presented in Table 2.

Table 2. The analysis results for an unreduced set of conditional attributes

Probability Pw	0.8	0.9
Number of elementary sets	4158	4158
Number of atomic useful rules	3076	2925
Relative probability of all atomic useful rules	0.733	0.702

The number of elementary sets in the table refers to the number of unique rules. An atomic useful rule is one that will occur at a probability higher than a predefined threshold P_w. Relative probability of atomic useful rules P_S is calculated as a sum of relative probabilities of all atomic useful rules divided by the number of elementary sets. The same parameters are generated by reducing successively the conditional attributes q_n to determine their significance Sq_n, according to the relation (X):

$$S_{qn} = \frac{P_{S\, for\, all\, attribute} - P_{S\, without\, qn}}{P_{S\, for\, all\, attribute}}$$

By soft reduction of subsequent conditional attributes in the course of the analysis we obtained significance levels of so reduced attributes. The results of this analysis for various levels of probabilities are presented in Tables 3 and 4.

Table 3. Significance analysis of conditional attributes with $P_w = 0.8$.

Conditional attribute	Number of reduced elementary sets	Number of reduced atomic useful rules	Relative probability of all reduced atomic useful rules	Attribute significance
r	2612	1676	0.63	0.13
w	2939	1876	0.63	0.14
lp	2260	1516	0.66	0.10
li	2728	1673	0.60	0.17
cw	2630	1610	0.60	0.18
md	2054	1380	0.66	0.10
z	2213	1484	0.66	0.10

Table 4. Significance analysis of conditional attributes with $P_w = 0.9$.

Conditional attribute	Number of reduced elementary sets	Number of reduced atomic useful rules	Relative probability of all reduced atomic useful rules	Attribute significance
r	2612	1537	0.59	0.16
w	2939	1711	0.58	0.17
lp	2260	1370	0.60	0.14
li	2728	1538	0.56	0.19
cw	2630	1475	0.56	0.20
md	2054	1256	0.61	0.12
z	2213	1360	0.61	0.12

The results of the analysis showed that the time of synchronization is the most important conditional attribute. Its significance at $P_w = 0.9$ reaches 20%, at $P_w = 0.9$ it also the most important. Such a high significance of this attribute results from the specific use of the Open MPI standard, where the results are sent through the network and synchronized at the end of calculations. An equally high level of significance is characteristic of the number of iterations *li*, whose significance roughly reaches 20%

for both values of P_w. The third significant attribute is communication time *w* (time of single datum transmission). Its significance reaches 17% and again is due to the characteristics of the Open MPI standard, which sends messages between nodes. The lowest significance level characterizes attributes of the number of calculation data for a thread *md* and the number of operations within the loop body *z*. Nevertheless, their level of significance above 12% is so large that these attributes are not reducible.

4 Conclusion

The paper presents two models for estimating the time of program loop execution in the parallel environment, compatible with Open MPI. The authors propose their own models of the execution time estimation for loops compatible with FAN and PAR transformations and verify model correctness. The models allow estimating the loop execution time on the basis of parameters characterizing the loops as such and the runtime environment used. This, in turn, allows using them in building efficient systems of task distribution in a parallel environment compatible with Open MPI. The significance analysis examines particular parameters used for building the time estimation models under consideration. The analysis showed a high significance of the individual attributes that should not be reduced. The inability to reduce these attributes shows that they have a high impact on the correctness of loop execution time estimation for loops running in an open parallel MPI, which indirectly affects the verification of the correct construction of the proposed model.

References

1. Rauber, T., Rünger, G.: Parallel Programming for Multicore and Cluster Systems, 2nd edn. Springer, Heidelberg (2012)
2. Abd-El-Barr, M., El-Rewini, H.: Fundamentals of Computer Organization and Architecture. Wiley, Hoboken (2005)
3. Pacheco, P.: An Introduction to Parallel Programming. Morgan Kaufmann Publishers, Elsevier, Burlington (2011)
4. Kshemkalyami, A.D., Singhal, M.: Distributed Computing, Principles, Algorithms, and Systems. Cambridge University Press, New York (2008)
5. Gramma, A., Karypis, G., Gupta, A.: Introduction to Parallel Computing, 3rd edn. Pearson, Upper Saddle River (2003)
6. Allen, R., Kennedy, K.: Optimizing Compilers for Modern Architectures A Dependence-based Approach. Morgan Kaufmann, Burlington (2001)
7. Pałkowski, M.: Algorytmy zwiększające ekstrakcję równoległości w pętlach programowych, praca doktorska. Politechnika Szczecińska, Szczecin (2008)
8. Lewis, T.: Foundations of Parallel Programming: A Machine-Independent Approach. IEEE Computer Society Press, Washington, D.C. (1992)
9. Wróbel, M.: Models for estimating the execution time of software loops in parallel and distributed systems. In: Theory and Engineering of Complex Systems and Dependability. Advances in Intelligent Systems and Computing, vol. 365, pp. 533–542 (2015)

10. NAS Parallel Benchmarks. http://www.nas.nasa.gov/publications/npb.html. Access date Sep 2016
11. Pawlak, Z.: Rough sets. Int. J. Comput. Inf. Sci. **11**, 341–356 (1982)
12. Ponce, J., Karahoca, A.: Data Mining and Knowledge Discovery in Real Life Applications. i-Tech Education and Publishing, Croatia (2009)
13. Xie, N.-X.: An algorithm on the parameter reduction of soft sets. Fuzzy Inf. Eng. **8**(2), 127–145 (2016). ISSN 1616-8658

On Application of Regime-Switching Models for Short-Term Traffic Flow Forecasting

Dmitry Pavlyuk(✉)

Transport and Telecommunication Institute, Lomonosova 1, Riga 1019, Latvia
Dmitry.Pavlyuk@tsi.lv

Abstract. This paper contributes to the identification of spatial dependency regimes in urban traffic flows. Importance of traffic flow regimes for forecasting and presence of spatial relationships between road network nodes are widely acknowledged both in traffic flow theory and empirical studies. In this research, we join these concepts and made the first steps to analysis of different regimes of spatial dependency in a traffic flow. Modern Markov-switching autoregressive distributed lag models are utilized and allowed to analyse the model structure in different traffic flow regimes. On the base of the models, we made a conclusion about the importance of traffic flow regimes for identification of a structure of spatial dependencies. The proposed approach is illustrated for real-world traffic flow data.

Keywords: Regime-switching model · Traffic forecasting · Autoregressive distributed lags · Spatial dependencies

1 Introduction

Since the beginning of the 19th century modelling and forecasting of traffic flows have become a point of many empirical and theoretical researches. Later a huge volume of available data that collected by loop detectors and growing computing power made it possible to apply modern mathematical models, in particular – time series analysis. Ahmed and Cook [1] applied a univariate autoregressive integrated moving average (ARIMA) model for urban traffic analysis in 1979, and hundreds of researchers utilized different time series techniques for traffic modelling and forecasting after that. Nowadays the applied traffic forecasting is moved to the big data era: a huge volume of traffic flow information with unclear hidden relationships and patterns is collected by multiple sensors and mobile devices. Data availability shifted the applied traffic flow analysis to multivariate spatial-temporal settings and lead to extensive utilization of modern models such as vector autoregressive (VAR) models, dynamic space-time models (STARIMA), among many other techniques [2–4].

This paper contributes to the identification of implicit regimes and spatial relationships, peculiar to traffic flow, using the advanced econometric techniques. An underlying assumption of this research is a dependency of traffic flow at a particular point from previous traffic flow characteristics in neighbour points, the regime of traffic flow and their interrelationships. Regime shifts have not received too much attention in the short-term traffic forecasting literature, and importance of regime identification for

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_33

forecasting accuracy is still arguable [5]. In its turn, importance of spatial relationships for traffic flow analysis recently has attracted scientific attention [6]. In this paper, we are making the first steps to merging these two concepts and develop a methodology of simultaneous identification of regime shirts and spatial dependencies and their dynamics.

2 Regime-Switching Models of Traffic Flow

Classical autoregressive models reconstruct a time pattern of the traffic flow and use this pattern for forecasting. Although the accuracy of such models is satisfactory for many practical purposes, it potentially could be improved by the utilizing the fact of different patterns of the traffic flow, which are switched over the time. For example, the traffic flow pattern could be different for free, stabilized or congested road conditions. Thus regime-switching models, which allow an existence of multiple dependency forms, became a promising technique for traffic flow analysis (models are referred as threshold models if time points of the regime switching are predefined and as Markov-switching (MS) models if the regime switching is a Markov process). Recently advanced regime-switching models were utilized by several researchers. Yu and Zhang [7] proved that the Markov-switching ARIMA outperforms regular ARIMA specifications in terms of in-sample prediction accuracy. Cetin and Comert [8] also found significant advantages of threshold models in case of presence time intervals with different traffic regimes. Kamarianakis et al. [9] found the threshold ARIMA model useful for discovering spatial dependencies between values of the traffic flow in different locations. Although the regime-switching techniques are promising, many researchers [5] are sceptical about their performance. Mainly the critics is based on core assumptions of regime-switching models – known time intervals for different regimes for threshold models and Markovian nature of regimes for MS models.

Another development direction of traffic modelling techniques is related to utilization of spatial relationships between different locations (sensors) for improving forecasting accuracy. A spatial structure is usually included in such models in a form of a predefined contiguity matrix, constructed on the base of physical road connections (i.e. STARIMA models [10]) or estimated statistically (using cross-correlation functions [6] or Granger causality [11]). Recently Min [3] utilized a dynamic STARIMA model that allows changing of the spatial contiguity matrix over time. In essence, this approach corresponds to continuous changes of a regime of the spatial dependency.

This paper is devoted to empirical research on an application of Markov-switching autoregressive models for discovering changes in spatial dependency regimes.

3 Research Models and Methods

The research methodology is mainly based on the autoregressive distributed lag model ARDL(p, q):

$$y_t = \sum\nolimits_{\mathrm{h}=1}^{\mathrm{p}} \alpha_{\mathrm{h}} y_{t-\mathrm{h}} + \sum\nolimits_{k=1}^{K} \sum\nolimits_{\mathrm{h}=0}^{\mathrm{q}} \gamma_{\mathrm{k,h}} \mathrm{x}_{\mathrm{k,t-h}} + \varepsilon_{\mathrm{t}}, \quad (1)$$

where t is a point of time (t = 0,…, T), y_t is a value of the forecasted stationary time series at t, $x_{k,t}$ is a value of the explanatory stationary time series k at t, p is a number of autoregressive lags, q is a number of lags of the explanatory time series, α_h, $\gamma_{k,h}$ are unknown coefficients, ε_t is a random disturbance at t. In this research, we will refer a model as “spatial” if it includes characteristics of the traffic flow at a neighbour location as explanatory time series.

We consider Markovian hidden states s that represent actual traffic flow regimes:

$$P\left(s_t | s_0, s_1, \ldots, s_{t-1}, y_0, y_1, \ldots, y_{t-1}, x_{k,0}, x_{k,1}, \ldots, x_{k,t-1}\right) = P\left(s_t | s_{t-1}\right). \quad (2)$$

Each regime has its own estimates of the ARDL, so the Markov-switching ARDL model is formulated as MS-ARDL(p, q):

$$y_{t|s_t} = \sum\nolimits_{\mathrm{h}=1}^{\mathrm{p}} \alpha_h(s_t) y_{t-\mathrm{h}} + \sum\nolimits_{k=1}^{K} \sum\nolimits_{\mathrm{h}=0}^{\mathrm{q}} \gamma_{\mathrm{k,h}}(s_t) \mathrm{x}_{\mathrm{k,t-h}} + \varepsilon_t^s, s_t = 1, \ldots, S. \quad (3)$$

Given a current regime, the model is reduced to the regular ARDL model.

A list of additional techniques, utilized in this research, includes:

- Augmented Dickey-Fuller test for stationarity,
- Cross-correlation function and the Granger causality test for identification of lagged relationships,
- Simple historical averages and exponential smoothing model for separation and long-term pattern and disturbances in the traffic flow,
- Akaike information criteria (AIC) and relative likelihood for model comparison.

4 Experimental Results

4.1 Data Description

We utilize data of traffic flows, publicly available from the Minnesota Department of Transportation. Data is collected by two sensors, S466 and S567, located before bridges in Minneapolis city centre and serving the same direction (Fig. 1). Selection of sensors arises from a goal of discovering implicit spatial relationships. Frequently researchers consider spatial relationships between nodes, located on the same road, so their significance directly follows from the physical nature of the traffic flow. For two selected sensors this relationship is not so straightforward: we assume that a share of drivers could choose one of two roads (bridges) at the point of junction on the base of their knowledge of recent flow densities on both roads. This information could be received from direct observations, but this is more likely that drivers just follow the recommendations from navigation software. Navigation software, in its turn, chooses a route on the base of recent traffic flow characteristics in both locations. If the described

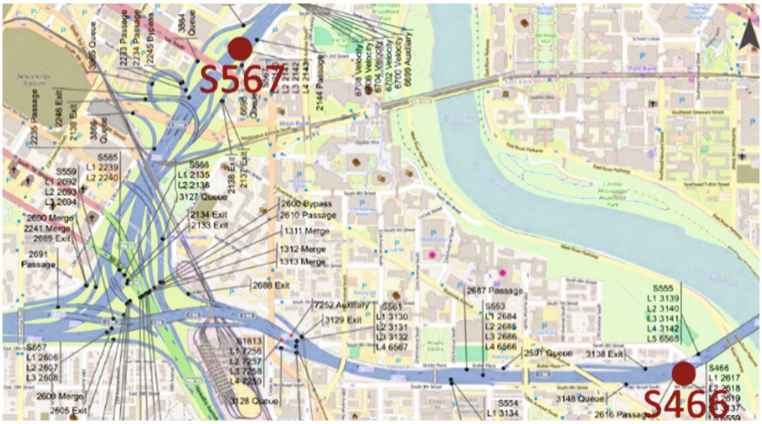


Fig. 1. Location of S466 and S567 sensors, Minneapolis, USA. Source: Minnesota Department of Transportation

scheme takes a place, a causal relationship between traffic flow characteristics at two selected locations will be identified.

Traffic information includes average speed and flow density values, aggregated by 5-minutes time frames. The research period is 2016/05/29-2016/09/03 (13 weeks). We concentrated on analysis of the speed-density relationship as one of the most well-researched theoretical traffic flow dependencies since Greenshield's linear model, published in 1935 [12]. Furthermore, the speed-density relationship is widely acknowledged as a multi-regime one [13], so the analyzed methodology should be appropriated. Also we selected the speed as the primary characteristic of interest, following the recently tendency to forecasting of travel times instead of traffic volumes [4]. Summary statistics for the research data set are presented in Table 1.

Daily patterns of speed and density values are presented on Fig. 2 and quite typical for urban traffic flows in city centre – small density with varying observed speeds at nights and congested traffic with stable speeds during daytime.

Both speed and density have clear historical profiles (or each time-of-day and day-of-the-week), which could be predicted by historical average values or seasonal

Table 1. Summary statistics

	S466		S567	
	Speed	Density	Speed	Density
Min	8.86	0.00	18.13	0.53
1st quarter	63.23	5.40	58.34	6.08
Median	66.63	13.28	60.85	16.16
Mean	65.38	13.27	60.82	14.60
3rd quarter	69.55	19.07	63.22	21.05
Max	120.00	102.43	92.43	143.55
Std. deviation	8.22	9.51	3.95	8.55

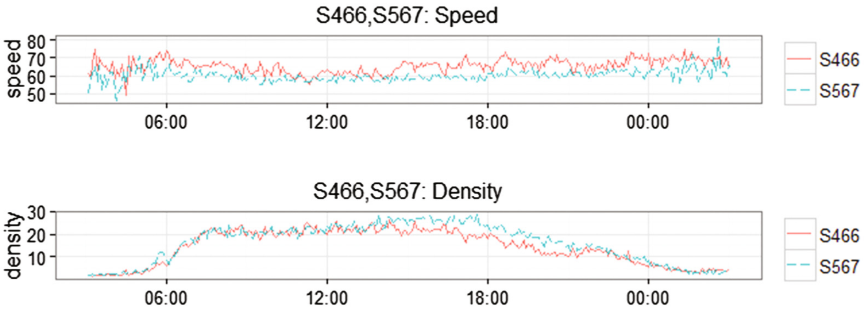


Fig. 2. Daily patterns of S466, S567 speed and density values

exponential smoothing (with weekly "seasons", s = 2016 for 5-minute time frames). The research period is relatively small and uniform, and exponential smoothing doesn't significantly outperform simple averages, thus we continue with the latter option:

$$Speed_t = Speed_{avg,t} + SpeedStat_t,$$

$$Density_t = Density_{avg,t} + DensityStat_t,$$

where $Speed_{avg,t}$ and $Density_{avg,t}$ are historical average values for day-of-the-week time-of-day, and $SpeedStat_t$ and $DensityStat_t$ are deviations from the historical averages and the main point of interest. Further these variables will be referred as stationarized.

SpeedStat and *DensityStat* values for both sensors are tested for stationarity using the augmented Dickey-Fuller test (lag order = 29) and all found stationary (results are presented in Table 2).

Table 2. Results of augmented Dickey-Fuller tests for stationarised speed and density values

	S466		S567	
	SpeedStat	DensityStat	SpeedStat	DensityStat
Dickey-Fuller	−21.76	−22.76	−19.55	−15.59
Sig.	0.002	<0.001	<0.001	<0.001
H_0: non-stationarity	Rejected	Rejected	Rejected	Rejected

4.2 Model Specification and Estimation

The main hypothesis of this research is formulated as: density of the traffic flow at the S466 point is a useful predictor for speed at the S567 point (probably, with the lag and with different regimes of dependency). A natural approach to identification of this relationship is a cross-correlation function. The plot of this function is presented on Fig. 3.

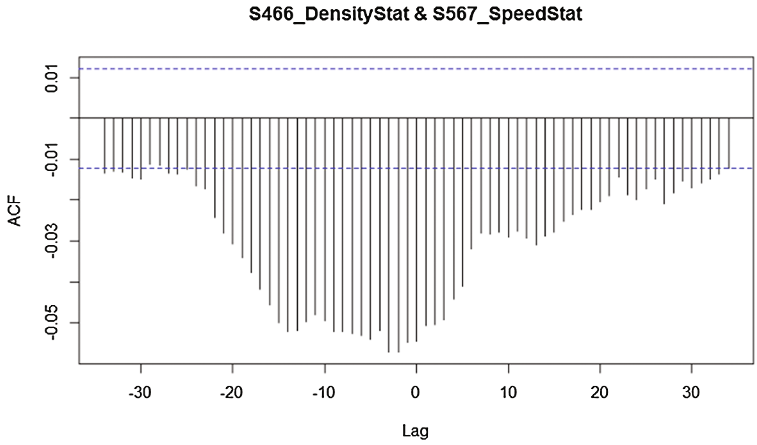


Fig. 3. Plot of cross-correlation of stationarised S466 density and S567 speed values

The plot has evidences of the relationship with a long memory (24 time points, 2 h). Although cross-correlation techniques are widely utilized in recent researches [6, 14], they are obviously related with a drawback for long-memory time series and the Granger causality test is more appropriate for discovering spatial relationships. Table 3 presents the results of the Granger causality test for stationarised speed and density values (lags = 16).

Among other significant spatial relationships, presented in Table 3, we note the significant causal relationship between DensityStat at the S466 point and SpeedStat at the S567 point, which provides an evidence for the main research hypothesis. All pairs of the research variables, except the pair S567: DensityStat - S466: SpeedStat, demonstrates the significant causal relationships. Since all variables are stationary, we selected the autoregressive distributed lag model for further analysis. So the stack of considered models includes:

Table 3. Results of the Granger causality test for stationarised speed and density values

Cause		Effect	F-statistic	p-value
S567: DensityStat	→	S567: SpeedStat	26.386	0.000
S567: SpeedStat	→	S567: DensityStat	11.933	0.000
S466: DensityStat	→	S466: SpeedStat	128.941	0.000
S466: SpeedStat	→	S466: DensityStat	4.201	0.000
S567: DensityStat	→	S466: SpeedStat	0.918	0.548
S567: SpeedStat	→	S466: SpeedStat	2.641	0.000
S567: SpeedStat	→	S466: DensityStat	1.007	0.446
S567: DensityStat	→	S466: DensityStat	2.275	0.003
S466: SpeedStat	→	S567: DensityStat	5.484	0.000
S466: DensityStat	→	S567: DensityStat	10.590	0.000
S466: SpeedStat	→	S567: SpeedStat	4.819	0.000
S466: DensityStat	→	**S567: SpeedStat**	**1.817**	**0.024**

- Autoregressive model AR(17), where 17 is the optimal number of speed autoregressive lags, selected on the base of the AIC criterion.
- Non-spatial autoregressive distributed lag model ARDL(17, 5), where the first parameter is defined as above and 5 is a number of density lags (all – at the S567 point).
- Spatial autoregressive distributed lag model ARDL(17, 5, 5), the first two parameters are defined as above and the third is a number of density lags at the S466 point.
- Markov-switching autoregressive distributed lag model MS-ARDL(17, 5) with 2, 3, and 4 regimes.
- Markov-switching spatial autoregressive distributed lag model MS-ARDL(17, 5, 5) with 2, 3, and 4 regimes.

Results of selected model estimation are presented in Table 4.

Table 4. Results of model parameter estimation

	AR(17)	ARDL (17, 5)	ARDL(17, 5, 5) spatial	2-regimes MS-ARDL(17, 5)		3-regimes MS-ARDL (17, 5, 5) spatial		
Regime				1 “night”	2 “day”	1 “peak”	2 “night”	3 “noon”
Intercept				−0.007	−0.021	0.015	−0.043	−0.024
s.e.				(0.036)	(0.173)	(0.022)	(0.059)	(0.014)
$SpeedStat_{S567,t-1}$	0.146	0.107	0.107	0.124	0.052	0.179	0.029	0.051
s.e.	(0.006)	(0.007)	(0.007)	(0.009)	(0.013)	(0.017)	(0.016)	(0.015)
$SpeedStat_{S567,t-2}$	0.124	0.097	0.096	0.091	0.072	0.114	0.057	0.035
s.e.	(0.006)	(0.007)	(0.007)	(0.008)	(0.013)	(0.019)	(0.014)	(0.013)
Higher lags of $SpeedStat_{S567}$ *are omitted in presentation*								
$DensityStat_{S567,t-1}$		**−0.152**	−0.003	**−0.038**	**−0.642**	−0.031	**−0.767**	**−0.048**
s.e.		**(0.011)**	(0.007)	**(0.007)**	**(0.054)**	(0.012)	**(0.046)**	**(0.011)**
$DensityStat_{S567,t-2}$		**−0.088**	−0.007	**−0.033**	−0.027	−0.020	−0.016	**−0.055**
s.e.		**(0.011)**	(0.009)	**(0.007)**	(0.062)	(0.013)	(0.026)	**(0.012)**
$DensityStat_{S567,t-3}$		**−0.048**	−0.009	−0.015	0.036	−0.010	0.040	−0.017
s.e.		**(0.011)**	(0.009)	(0.007)	(0.012)	(0.009)	(0.046)	(0.013)
Higher insignificant lags of $DensityStat_{S567}$ *are omitted in presentation*								
$DensityStat_{S466,t-1}$			**−0.151**			−0.012	−0.022	0.009
s.e.			**(0.011)**			(0.010)	(0.024)	(0.005)
$DensityStat_{S466,t-2}$			**−0.088**			0.002	−0.062	−0.001
s.e.			**(0.011)**			(0.001)	(0.029)	(0.001)
$DensityStat_{S466,t-3}$			**−0.048**			−0.016	0.014	−0.014
s.e.			**(0.011)**			(0.020)	(0.018)	(0.009)
Higher insignificant lags of $DensityStat_{S466}$ *are omitted in presentation*								
AIC	150784	150382	150385	138286		137639		

The univariate AR(17) (which is a practical equivalent in terms of goodness-of-fit of ARMA(3, 2) for this sample) and ARDL(17, 5) are used as a base for model comparison. It should be noted that inclusion of non-spatial density lags into the model significantly increases its relative quality in terms of AIC.

Spatial ARDL(17, 5, 5) model, which also includes density lags at the neighbour point S466, is worse than the non-spatial ARDL(17, 5) model (significance of spatial lags can be explained with a relationship between densities in neighbour points, discovered above). So up to this moment we conclude the absence of spatial relationships, despite the results of the cross-correlation function and the Granger causality test, discussed above.

The next part of the research hypothesis is related to the existence of the different regimes of flows, where the spatial dependence can take a place. We analysed Markov-switching ARDL models with 2, 3, and 4 regimes and selected the model with 2 regimes as the optimal non-spatial model and the model with 3 regimes as the optimal spatial model. Daily probabilities of the regimes in 2- and 3-regime models are presented on Fig. 4.

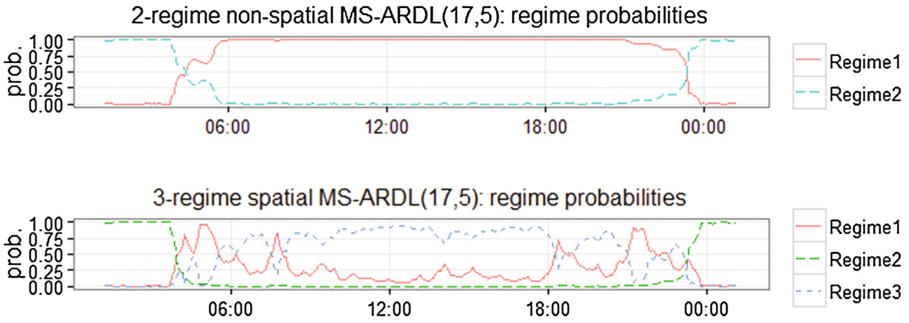


Fig. 4. Daily pattern of estimated regime probabilities in MS-ARDL models

Regimes in the 2-regime MS-ARDL model clearly correspond to night (Regime2) and day (Regime1) hours. The existence of this pattern doesn't correspond to the basic assumption of Markov regime switching, so the model shouldn't be used for out-of-sample prediction. Instead, we recommend to use the discovered regimes in threshold model definition. Generally, this approach (identification of thresholds using MS models and their further usage in a threshold model specification for forecasting) looks perspective and require additional research.

The spatial 3-regime MS-ARDL model is slightly better than non-spatial in terms of AIC. Regimes in this model are also explainable – night hours (Regime2), stabilized noon hours (Regime3) and morning-evening transitional hours (Regime1). The main point of interest of this research lies in analysis of spatial dependencies in different regimes. Although the dependency between speed at the S567 point and density at the S466 point is found significant in the ARDL model without regimes, the MS models with 2, 3 or 4 don't provide significant evidences for this hypothesis. The experimental results are too limited for a general conclusion, but at least we recommend that researchers pay attention to different regimes before investigating spatial relationships in a traffic flow. Analysis of the spatial dependency dynamics is a promising research direction both from theoretical and applied perspectives.

5 Conclusions

This paper contributes into the statistical identification of regimes and spatial dependencies in short-term traffic modelling and forecasting. We utilized regime-switching modifications of classical autoregressive distributed lag models to discover different regimes of traffic flow and spatial dependencies, peculiar to every regime.

- Literature on different regimes of spatial dependency in traffic flow is very limited. Also we noted the growing research interest to the identification of regimes and spatial dependencies for improving an accuracy of short-term traffic forecasting.
- Classical techniques that don't take switching of regimes into account (such as cross-correlation and Granger causality) can lead to incorrect conclusions about spatial dependencies in traffic flows.
- Markov-switching autoregressive models can be used for identification of time intervals with different regimes. Further these intervals (or their forecasts) can be used in threshold models for out-of-sample forecasting.
- Although the experimental results of this research don't provide evidences of different regimes of spatial dependencies with two road nodes, the proposed methodology allows such regime identification and require additional validation.

References

1. Ahmed, M.S., Cook, A.R.: Analysis of freeway traffic time series data by using Box-Jenkins techniques. Transp. Res. Rec. **722**, 1–9 (1979)
2. Kamarianakis, Y., Prastacos, P.: Forecasting traffic flow conditions in an urban network: comparison of multivariate and univariate approaches. Transp. Res. Rec.: J. Transp. Res. Board **1857**, 74–84 (2003)
3. Min, X., Hu, J., Chen, Q., Zhang, T., Zhang, Y.: Short-term traffic flow forecasting of urban network based on dynamic STARIMA model. In: 2009 12th International IEEE Conference on Intelligent Transportation Systems, pp. 1–6 (2009)
4. Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C.: Short-term traffic forecasting: where we are and where we're going. Transp. Res. Part C: Emerg. Technol. **43**, 3–19 (2014)
5. Williams, B.M.: "Real-time road traffic forecasting using regime-switching space-time models and adaptive lasso" by Y. Kamarianakis, W. Shen, and L. Wynter. Appl. Stoch. Models Bus. Ind. **28**, 319–321 (2012)
6. Ermagun, A.: Network Econometrics and Traffic Flow Analysis (2016). http://conservancy.umn.edu/handle/11299/183378
7. Yu, G., Zhang, C.: Switching ARIMA model based forecasting for traffic flow. In: 2004 Proceedings on Acoustics, Speech, and Signal Processing, vol. 2, pp. 429–432. IEEE (2004)
8. Cetin, M., Comert, G.: Short-term traffic flow prediction with regime switching models. Transp. Res. Rec.: J. Transp. Res. Board **1965**, 23–31 (2006)
9. Kamarianakis, Y., Shen, W., Wynter, L.: Real-time road traffic forecasting using regime-switching space-time models and adaptive LASSO. Appl. Stochast. Models Bus. Ind. **28**, 297–315 (2012)
10. Kamarianakis, Y., Prastacos, P.: Space–time modeling of traffic flow. Comput. Geosci. **31**, 119–133 (2005)

11. Li, L., Su, X., Wang, Y., Lin, Y., Li, Z., Li, Y.: Robust causal dependence mining in big data network and its application to traffic flow predictions. Transp. Res. Part C: Emerg. Technol. **58**, 292–307 (2015)
12. Greenshields, B.D.: A study of traffic capacity. In: Proceedings of the Highway Research Board, pp. 448–477 (1935)
13. Sun, L., Zhou, J.: Development of multiregime speed-density relationships by cluster analysis. Transp. Res. Rec.: J. Transp. Res. Board **1934**, 64–71 (2005)
14. Yang, S., Shi, S., Hu, X., Wang, M.: Spatiotemporal context awareness for urban traffic modeling and prediction: sparse representation based variable selection. PLoS ONE **10**, e0141223 (2015)

Critical Information Infrastructure Protection Model and Methodology, Based on National and NATO Study

Lachezar Petrov[1], Nikolai Stoianov[2(✉)], and Todor Tagarev[2]

[1] Ministry of Defence of the Republic of Bulgaria, CIS Directorate,
Policy and CIS Development Planning Branch, Sofia, Bulgaria
l.t.petrov@mod.bg
[2] Bulgarian Defence Institute,
2 Prof. Tsvetan Lazarov Blvd, 1592 Sofia, Bulgaria
{n.stoianov, t.tagarev}@di.mod.bg, nik.stnv@gmail.com

Abstract. National and international security, our financial, industrial as well as economic prosperity, healthcare system and national well-being as a whole are dependent on critical infrastructures, which could be described as highly interdependent. Many examples are available such as the national electrical grid, oil and natural gas systems, telecommunication and information networks, transportation networks, water systems, and banking and financial systems. Keeping them in reliable and secure state and study their dependencies is paramount for every government or organization. There is an urgent need of their classification. Creation and development of model and methodology which could describe their behaviors is going to make this world safer. The presented here model and based on it study and initial results are steps toward reliable and secure critical information infrastructure.

Keywords: Cyber security · Critical information infrastructure · Cyber model

1 Introduction

Interrelations in most of the aspects of our modern life are based on a complex system of complementary and in some cases mutually exclusive network connections that organize contacts to multiple system interfaces. So the realization of any product or any task becomes dependent on large and complex, but relatively logical mechanism based on infrastructure and serviced by communication and information systems and networks from different class. Such a complex mechanism, enables the organizations (public, private, non-governmental etc.) to conduct their interests and to achieve their goals. So the meaning of "security" and the ability to achieve it requires a different approach from the conventional.

The term "cyber defense" becomes extremely important, especially after the NATO summit in Warsaw in 2016, when "cyber space domain" [1] was established as a new operational domain, in addition to the existing Air, Sea and Land domains. The emergence of a new area in military and economic confrontation defines the necessity of developing a new scientific area and cooperation on a different level and variety of

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_34

requirements. The last few years have clearly demonstrated that this approach is right, particularly in a situation of hybrid threat and use of the opposing sides of the so-called "soft power" to resolve any problems in a wide range of areas. Taking into account how dynamic is the development of the technology, concepts named "cyber defence" and "cyber space domain" acquires particular significance in terms of the critical infrastructure of any type.

Although NATO has adopted the term "cyber space domain", it is doubtful that critical infrastructure of military or civilian type could be divided, as far as those terms have largely lost their identity and are strongly intertwined, with no possibility to be set independently.

2 Infrastructure vs. Critical Infrastructure

To determine the scope of what the authors would like to achieve, it is necessary to define the terms that we will handle. The term "infrastructure" had been introduced in the nineteenth century by Swiss military theorist Antoine-Henri Jomini, who highlights its strategic and operational importance for the leadership during any military operation [2]. Purely military use of the term has been lasted until the mid-twentieth century. At this time the term had indicated the territorial organization of an army battlefield. After the middle of this century, the term "infrastructure" had been recognized and began to be used in economics and management theory. Now the term is used in almost every field of science and is very common in studies related to security.

In our opinion there is still no comprehensive and widely accepted definition of the term "critical infrastructure" This fact is only an illustration of the importance of the critical infrastructure and the interests, connected to it. Many researchers in various fields have defined the term "critical infrastructure" but for the means of this article the authors will stick to a definition, synthesized from various sources.

Critical infrastructure is a system of facilities, services, rules, personnel, documents, management methodology and procedures of processing and exchanging information, whose malfunction or destruction for whatever reason, would have a serious negative impact on the health and safety of people and environment as well as could lead to serious financial and material losses and would violate the effective functioning of the state and/or military governance in any region or country.

In general, the basic types of the critical infrastructure subdivision are described on Fig. 1. [3]

Increasingly clear is the tendency to change the center of gravity of threats from physical or purely military impact (conventional military conflict) to indirect/not conventional impact on the enemy's critical infrastructure elements [4].

It is obviously how important and contemporary the terms "terrorism", "cyberwar" and "environmental changes" are, often grouped under the general term "hybrid war". The primary purpose of "hybrid war" approach is not to be destroyed the enemy's critical infrastructure. The main purposes is, the violation of the enemy's critical infrastructure work to cause crises, through which the critical infra-structure to be unbalanced. The main advantage of this approach is the possibility at a later stage, this critical infra-structure easily to be recovered and managed [5].

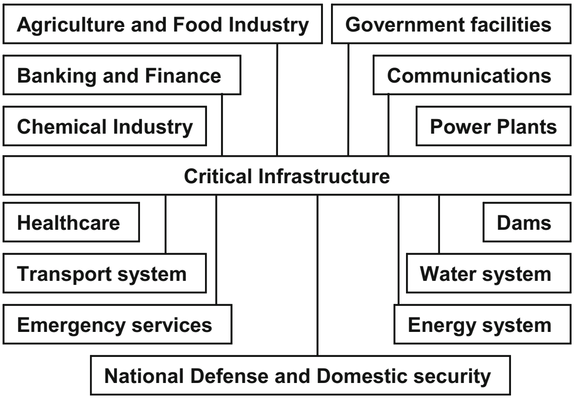


Fig. 1. Main types of critical infrastructure

Multilateral features of the term "critical infrastructure" determines as an optimal the following fragmented classification [6]:

- In accordance with the location:
  - Critical terrestrial objects - sites located throughout the country;
  - Critical marine sites - ships, oil and gas marine platforms, pipelines.
- In accordance with the mobility:
  - Fix/stationary - manufacturing equipment, power plants, transportation facilities (airports, ports, oil, gas and fuel facilities, rail, bus and marine stations), underground special equipment – mainly communications, control stations, warehouses of state raw materials and fuel reserves, laboratories, etc.;
  - Mobile - aircraft, ships, ground transportation, even satellite communication means;
- In accordance with their public role and social significance:
  - Administrative buildings - district and other smaller centers and municipalities;
  - Objects of the energy system;
  - Sites of the chemical industry, working with hazardous and toxic substances;
  - Objects, parts of the transport system of any kind - ports, airports; marine, railway and bus stations, highways and shuttle lines (roads), bridges and passages.
  - Sites of domestic security such as drinking water system; food establishments (grain warehouses, oil mills, bread factories, meat and dairy farms, wineries, hypermarkets);
  - Polyclinics and hospitals, universities and schools; resorts; buildings and complexes for socio-economic, commercial and entertainment activities, business forums, theaters, sports and other festivals and competitions facilities with great daily and seasonal attendance.
- Objects of CI with year-round importance;
  - Objects of CI with seasonal importance - the seasonal nature is formed in the summer of short-term concentration of huge mass of tourists mainly in urban settlements and resorts of southern European Mediterranean type along the coast as well as in the country site (Fig. 2).

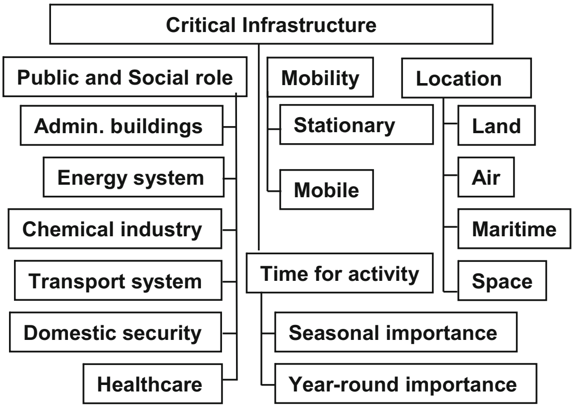


Fig. 2. Critical infrastructure classification

It is important to be mentioned that such a classification and a list of objects is not and cannot be finalized. It is continuously subject to adjustment and expansion, especially in this dynamic security environment and variety of public relations.

If we focus on the types of interdependencies of the critical infrastructure, we can define four classes:

1. Physical dependence - physical dependences that comes from physical connections or links among different elements of the infrastructure. In this context, interruptions and disturbances in one infrastructure can spread to other infrastructure projects.
2. Cyber dependence - interdependencies that occur when the infrastructure is dependent on information transmitted through the information infrastructure. Such relationships are the result of increased use of computer-based information systems that support surveillance and management activities.
3. Geographical dependence – dependence, which exists between the two infrastructures, when a local environment event can cause problems in both. This usually happens when the infrastructure elements are in close spatial proximity;
4. Logical dependence - relationship that covers all dependencies far from all above mentioned, which could be caused for example by regulation, legal or political restrictions.

The four described types of relationships are not mutually exclusive, although each has its own characteristics [3].

Additional complexity in terms of interdependencies occurs, when the information infrastructure is categorized by two different key dimensions:

- Service-oriented interconnections;
- Information and data oriented interconnections.

Although the integration of critical infrastructures and the synergy in its usage undoubtedly provides valuable benefits in terms of efficiency, service quality and cost reduce, the interdependencies increase the vulnerability of the critical infrastructure, as they lead to avalanche effect in distribution of errors from one critical infrastructure of another. This creates problems, whether their exposure is accidental or effect of malicious threat. Even a simple power outage caused by a problem, mismanagement or operator intervention is able to lead to cascading outages and ultimately to the collapse of the whole system. There are many examples of cascading problems of infrastructure dependencies that lead to catastrophic events in multiple infrastructures that can cover wide geographical areas.

Organizational and economic logic stimulates the usage of Internet as well as globalization enables to different organizations, wherever they are located, to work as a whole. Communication technologies improve productivity, efficiency and competitiveness. Nowadays many organizations (both governmental, industrial and financial as well as military) focus much of its activity by consolidating the operations through virtual tunnels to a central location for processing of all data. In such cases, Internet use reduces operating costs. With the increasing number of transactions, huge amounts of data with varying degrees of protection flow and pass through the Internet.

The society has evolved to the state of dependency on the availability, reliability, safety and security of main infrastructures of any type. It is due to significant social and economic benefits they provide. Unfortunately, in case of malfunctioning or improper protection appear extremely serious negative consequences due to the fact that all systems have become a necessity. The gradual introduction of total management of all networks, the introduction of systems for monitoring and control as well as the interdependence that always arise in cases like this, certainly optimize and improve the level of performance in the critical infrastructure. Along with the benefits, such approach permits access of cyber criminals and terrorists, with all negative consequents. So the scenario becomes more complex, based on the fact that the modern technologies introduce new sources of potential risk, upon the traditional threats.

Even the broad defined above classification of critical infrastructure presents its diversity. This creates even greater variety of parameters that would have to be taken into account in determining the criteria how critical one infrastructure could be. In order to optimize efforts to determine adequate measures in case of problems with critical infrastructure, a researcher has to focus on a specific segment of this very broad concept. During the development of this material a focal point was mainly a segment named “Critical Information Infrastructure” - CII.

3 Methodology and Enhanced Methodology

The main objectives that an organization has to follow preparing actions for critical information infrastructure protection could be specified as follows [7]:

1. Determination of critical information infrastructure at national level;
2. Preparing the methodology and conducting a national survey to determine the dependencies of critical information infrastructure of information systems involved in the management of the state;
3. Development of a national program to protect critical information infrastructure;

4. Development of rules and standard operational procedures to assist owners and operators of critical information infrastructure (both government and private) to minimize the risk of the collapse of parts or whole segments thereof;
5. Definition and description of problems with cross-sector dependencies;
6. Development of policies and standard operational procedures all together with the International Critical Infrastructure Protection (CIP)/Critical Information Infrastructure Protection (CIIP) organizations setting a transnational solutions and minimize the consequences;
7. Control and measuring the level of maturity achieved in CIP/CIIP and following procedures of adjustments the legislation, strategies, rules and procedures based on the results.

To limit somehow the scope of this paper, the authors will focus primarily on the first two points of the proposed above plan for the protection of CIIs. There are various algorithms based on a set of parameters that could determine whether an infrastructure is within the scope of the definition “critical”. Each national unit for Cyber Security/Defense either defines his own algorithm or adjusts one by adding any relevant national parameters to already developed and verified algorithms of other organizations.

Based on the theory mentioned before, the authors conducted a study, during which, national critical information infrastructure had been designated and a set of critical dependencies had been defined, based on critical tasks, performed by NATO in support international peace and stability [8].

Beginning of the study had been given by determination of the entire national critical infrastructure working in favor of the state management and based on the criteria set up in Fig. 1.

On this basis and following the steps, described in the methodology, we had determined as well the national information infrastructure which manages and control the national critical infrastructure. Following the objectives of the study the authors designated a summary list with 256 critical tasks that NATO performed. It had been found, that our national information infrastructure is an important factor for 49 of these critical tasks.

Subsequently NATO critical tasks execution had been compared with the tasks, performed by national information infrastructure. Based on this comparison the authors discovered that 66% of defined national information infrastructure falls in the scope of the definition “critical information infrastructures”.

On such designated national critical information infrastructure works two information platforms serving a total of nineteen IT applications. They all could be described with the definition, given at the beginning of the material.

From Table 1, it is visible, that all described infrastructure and systems are critical from pure national point of view, but just any of them are critical from NATO prospective. Part of these infrastructures are important and valuable for the collective defense but they are not critical [9].

Table 1. Critical information infrastructures involvement in National and NATO critical tasks performance

Critical information infrastructures	National critical tasks involvement	NATO critical tasks involvement
National defense information infrastructure	x	x
Domestic security information infrastructure	x	x
National road management information system	x	x
National railway information infrastructure	x	x
Information infrastructure of BG National Bank	x	
Water supply management information infrastructure	x	
Gas management information system	x	
Ministry of Foreign affairs information system	x	x
Judicial information system	x	x

4 Summary and Further Recommendations

At the end it should be noted that the determination of national critical information infrastructure is not an end in itself. This process allows the determination of priorities in the development of national programs for the protection of CII. This also helps in development of national policies and standard operational procedures to assist owners and operators of CII (both government and private) in order to minimize the risk of failure. Such approach gives us an idea for the steps that should be taken in order an integrated security system to be created.

As a conclusion I would like to underline, that the results from the study, based on the above described model and methodology, are reliable enough and the figures are acceptable. Such study could be reported as a very close to the reality, in case that cross border connectivity is included. Following such course of action is going to increase the number of parameters and variables, but it will make the model and methodology more mature.

The recommendations for the future enhancement – the model and methodology could be used easily and the reliability will increase, after the development of special software, deliberately constructed for testing critical information infrastructures, with possibility many variables and parameters to be object of modification. This will allow one information infrastructure to be tested with variable of parameters and suitable and secure protection model to be followed.

References

1. North Atlantic Treaty Organization NATO - NATO's capabilities. http://www.nato.int/cps/en/natohq/topics_49137.htm?selectedLocale=en. Accessed Jan 2017
2. Antoine-Henri (Baron) de Jomini: The art of war, Arc Manor, Library of congress control number 2006936549. http://www.arcmanor.com/FDL/AofW5674.pdf. Accessed Jan 2017
3. Rinaldi, S.M.: Modeling and simulating critical infrastructures and their interdependencies. In: 37th Annual Hawaii International Conference on System Sciences, 8 p. IEEE Conference Publications (2004). doi:10.1109/HICSS.2004.1265180
4. Echevarria II, L.C.A.J.: Clausewitz's center of gravity, its not what we thought. Naval War Coll. Rev. **LVI**(1), 108–123 (2003)
5. Vatis, M.A.: Cyber Attacks during the War on Terrorism: A Predictive Analysis. Dartmouth Coll Hanover NH Inst for Security, Accession Number: ADA395300
6. Homeland Security WEB Page, Critical Infrastructure Sectors. https://www.dhs.gov/critical-infrastructure-sectors. Accessed Jan 2017
7. ENISA, Europa. https://resilience.enisa.europa.eu/enisas-ncss-project/CIIPApproachesNCSS.pdf. Critical Information Infrastructures Protection approaches in EU, Final Document, Version 1, TLP: Green, July 2015
8. Bulgaria National Security and Defence Strategies. https://www.md.government.bg/bg/doc/drugi/20101130_WP_BG.pdf. White Paper on Defence and the Armed Forces of the Republic of Bulgaria (2010)
9. National cyber security strategy "Cyber Resilient Bulgaria 2020" (2016). http://www.cyberbg.eu/

The Method of Creating Players in the Marketing Strategy

Henryk Piech, Aleksandra Ptak, and Michal Saczek(✉)

Czestochowa University of Technology, Dabrowskiego 69, Poland
h.piech@adm.pcz.czest.pl, olaptak@zim.pcz.pl,
michal.saczek@icis.pcz.pl

Abstract. Internet marketing takes a variety of forms and uses different formalisms and grammar [1]. In online stores, entering the keyword we receive harmoniously arranged (on the web page) offers from a given domain as well as related and quite distinct domains [2]. The greatest challenge for marketing system designers is a choice of appropriate variants of offers arrangement in terms of diversifying their types. Another significant issue is also the distribution, which is, for example, the frequency of use of particular variants of offer pages. The aim of our research is to predict client's needs and preferences in more accurate way. The strategy that we have chosen is supposed to increase the attractiveness of an offer, which indirectly facilitates customers' product searches and affects his or her purchase decision. Common product-offering models are extremely poor in functions and they do not allow us to determine any factors that would decide on objects which are more likely to meet expectations of a client. In those models related products of chosen category are selected randomly what is ineffective. Due to that we are going to improve it and make the access to desired products easier and quicker for customer by building an offer based on characteristics referring to demand, sales volume, trends, etc.

The above issues suggest the possibility of using strategic games in which the players are variants (strategies of model structuring) of product-offering solutions applied with a particular probability distribution (frequency of using product-offering models).

Strategies are also connected with the assessments of their effectiveness, that is player's payoffs which will not be described in this article. The main focus of this work is the idea of creating players by determining their payoff [9]. In order to explain this concept precisely the authors present the structure containing formalisms (Sect. 1), the general description of the proposal of creating competing models (Sects. 2, 3), the example (Sect. 4) and summary (Sect. 5).

Keywords: Game theory · Random distribution · Entropy · Marketing strategy

1 Introduction

Selection of a system base is based on an existing offering solution containing graphic - descriptive blocks relating to products of different assortment (selected domain **D(s)**) and various subclasses (subdomains $\mathbf{PD_i(s)}$) supplementing the usage and possible

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_35

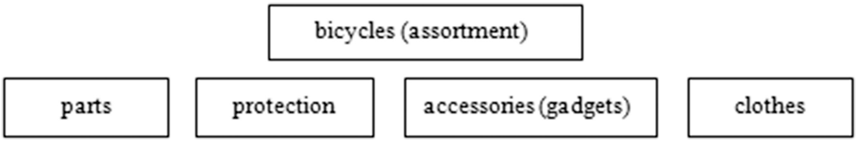


Fig. 1. Domain and subdomains of sale offers.

applications of the selected offer (domain and subdomains) [4]. Exemplary, it is illustrated in Fig. 1.

Further connections may refer to related domains (**DP$_i$(s)**). Relation is a conventional thing but it should be based on generally accepted principles, trends and perceptions, as it is addressed to the average (we refer to the most numerous) customer. Exemplary, it is illustrated in Fig. 2.

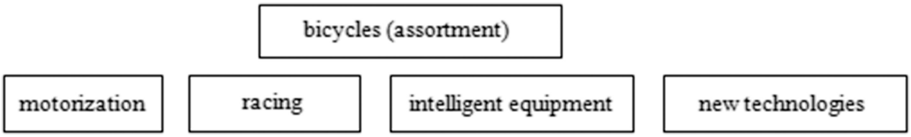


Fig. 2. Domain and related domains in the sales offer.

The project authors themselves have to decide upon the application and domains relation scale, for example, based on current trends, including environmental trends.

Another strategy is the inclusion of blocks of domains completely detached from goods chosen by the customer (**OD$_i$(s)**). The bidder at the time is guided by a market priority, shaping an interest (directing the interest to other objects) or changes in customer preference (Fig. 3).

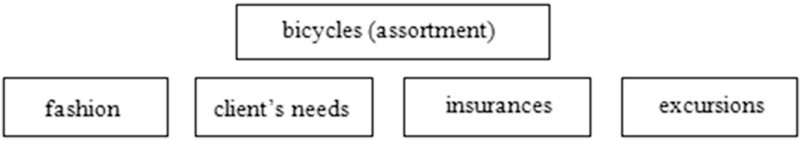


Fig. 3. Domain and domains in the lesser extent or entirely unrelated to the choice of the customer

For completeness of the formalisms let us create the grammar related to the relation and operators. The relation domain - subdomain can be described by the implications operator, as the subdomain is an inherent attribute of the domain:

$$\mathbf{D}^{(s)} \; = \; > \; \{\mathbf{PD}_{\mathbf{i}}^{(s)}\}\,,\; \mathbf{1} \le \mathbf{i} \le \mathbf{l}_{\mathbf{pd}}, \tag{1}$$

where
i, $\mathbf{l_{pd}}$ – the number of the subdomain and the number of all subdomains,
"=>" – implication referring to a strong relation.

Domains relation can be understood as the link weaker than inherent and the authors suggest to be marked as follows:

$$\mathbf{D}^{(s)} \cong \; > \; \{\mathbf{DP}_{\mathbf{i}}^{(s)}\}\,,\; \mathbf{1} \le \mathbf{i} \le \mathbf{l}_{\mathbf{pd}} \tag{2}$$

where
i, $\mathbf{l_{dp}}$ – the number of related subdomain and the number of all related subdomains,
"≅>" – dependence corresponding to a weak relation.

That brings us to unrelated domains. Similarly, the authors suggest the designation:

$$\mathbf{D}^{(s)} \approx \; > \; \left\{\mathbf{OD}_{\mathbf{i}}^{(s)}\right\},\; \mathbf{1} \le \mathbf{i} \le \mathbf{l}_{\mathbf{od}}, \tag{3}$$

where
i, $\mathbf{l_{od}}$ – the number of unrelated subdomain and the number of all unrelated subdomains,
"$\approx$ >"– marking irrelevant or accidental dependence.

For the description of the suggested method of creating players in the strategic analysis one needs to introduce the concept of the domain basic objects (the basic elements, meaning products offered $\mathbf{P}_{\mathbf{j}}^{(\mathbf{i})}$ – **j**-th product of **i**-th domain) because only they will be offered to the customer. Therefore one can formally recommend following notation of number of dependencies:

$$\mathbf{D}_{\mathbf{i}} \; = \; \bigcup_{\mathbf{j}=\mathbf{1}}^{\mathbf{lp}(\mathbf{i})} \mathbf{P}_{\mathbf{j}}^{(\mathbf{i})} \text{ or } \mathbf{D}_{\mathbf{i}} \; = \; \{\mathbf{P}_{\mathbf{j}}^{(\mathbf{i})}\},\; \mathbf{1} \le \mathbf{j} \le \mathbf{lp}(\mathbf{i}), \tag{4}$$

where
lp(i) – the number of products in terms of **i-**th domain.
Supplemental relations are less useful to describe the proposed method, as they relate to the definition of inseparability, relation and lack of it:

inseparability:
$$\exists \mathbf{k}\; \{\mathbf{D}^{(\mathbf{i})} \bigcap \mathbf{D}^{(\mathbf{j})} = \mathbf{P}_{\mathbf{k}} \neq \varnothing\}$$

relation (on the base of the intermediary domain $\mathbf{D}^{(\mathbf{v})}$):
$$\exists \mathbf{D}^{(\mathbf{v})}\; \{(\mathbf{D}^{(\mathbf{i})} \bigcap \mathbf{D}^{(\mathbf{v})} \neq \varnothing) \wedge (\mathbf{D}^{(\mathbf{j})} \bigcap \mathbf{D}^{(\mathbf{v})} \neq \varnothing)\}$$

$$\begin{aligned}&\textbf{dissimilarity}\ (\textbf{lack of relation}):\\&(\neg\exists \mathbf{k}\ \{\mathbf{D}^{(\mathbf{i})} \bigcap \mathbf{D}^{(\mathbf{j})} = \mathbf{P}_{\mathbf{k}} \neq\ \varnothing\})\\&\wedge\\&(\neg\exists \mathbf{D}^{(\mathbf{v})}\ \{(\mathbf{D}^{(\mathbf{i})} \bigcap \mathbf{D}^{(\mathbf{v})} \neq \varnothing) \wedge (\boldsymbol{D}^{(j)} \bigcap \boldsymbol{D}^{(v)} \neq \varnothing)\})\end{aligned}$$

The above definitions can have quite different character, which does not affect the presentation of the concept and methods of creating strategic players.

2 Structural Limitations in Offers - Dynamic Regulations of the Probability Density of an Advertisement Elements Selection

Limitations of space in the structure of advertisement are arising from the principles of perception and concentration of customers. The offer should be short and interesting, which results in reaching the customer, understanding the resulting potential benefits and ultimately buying the product [5]. Harmonious structure should contain different types of products in a certain way, interconnected or attractive in another way. Mutual relations between domains are the basis for differentiating the frequency of use of different structural offers models.

The authors propose to define models of offers by quantitative thresholds range of use associated (or not) domains. These thresholds will limit the number of products with randomly matched domains (selection will be made on the basis of the distribution set and updated after stages). Products from the chosen field will also be selected randomly, but with a given probability distribution. The authors introduce therefore proposed formalisms:

ThD – the threshold of number of products from domain selected by the customer **(s)**,
ThPD – the threshold of number of products from randomly selected subdomain according to the given distribution j,
ThDP – the threshold of number of products from randomly selected related domain according to the given distribution **k**,
ThOD – the threshold of number of products from randomly selected subdomain according to the given distribution **u**.
The strategic model will be therefore defined by the following four **{ThD, ThPD, ThDP, ThOD}**.

To determine the frequency of use of alternative different models, the authors use the analysis used in the strategy games. Models will therefore play the role of players. Games will be played with competition but also between their own places and moments of activation from different offers models. Let us return to the principle of generation, which is based on J. von Neumann's algorithm of elimination [6], supplemented by the authors by multiplying the elimination stages (four-dimensional random structures) and after stages corrections of the probability density. The offer structure can be thus shown as follows (Fig. 4). Products and their descriptions are randomly selected by a given

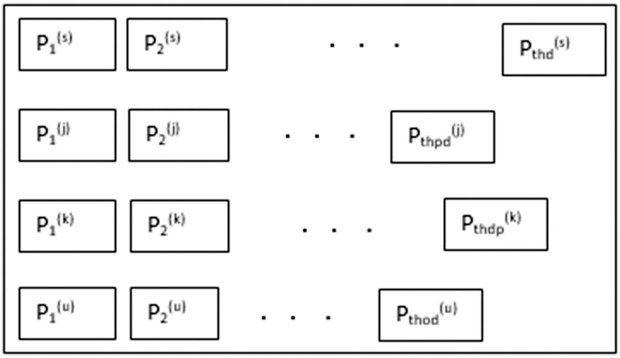
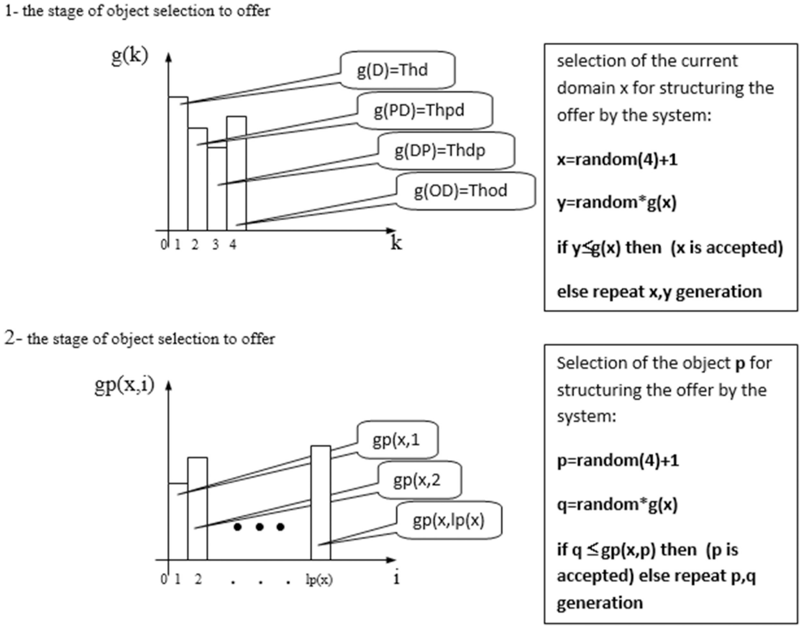


Fig. 4. Structure of the offer presentation


1- the stage of object selection to offer

selection of the current domain x for structuring the offer by the system:

x=random(4)+1

y=random*g(x)

if y≤g(x) then (x is accepted)

else repeat x,y generation

2- the stage of object selection to offer

Selection of the object **p** for structuring the offer by the system:

p=random(4)+1

q=random*g(x)

if q ≤gp(x,p) then (p is accepted) else repeat p,q generation

Fig. 5. Stages of the object offer generation (the boxes include description of the algorithm of random code generation with distribution determined by using von Neuman method).

distribution, whose form is determined empirically in two stages (Fig. 5). Objects selections are limited by the given thresholds, which gives the opportunity to build a criterion for termination of the offer structuring [7].

3 Algorithmization of the Dynamic Adjustment (Correction) of the Probability Density Function

Density function for domains in different methodological interpretations depends on the sale level, products prices, clients interest or, for instance, from the level of given information. The amount of information is being determined by the amount of entropy [8]. It is also possible to link the price with the level of transferred information in the offer. This concerns mainly the domains **PD**, **DP**, **OD**. In the domain chosen by the customer $D^{(3)}$ to create density function one only uses the size of the given thresholds **ThD**, **ThPD**, **ThDh**, **ThOD**.

Moving on to adjustments of the density function one can extract main assumptions:

(a) The choice of the one of the basic domains in the process of the offer creating (Fig. 5: 1- stage), leading to the positive verification of the elimination conditions, that is, to find product **p** (Fig. 5: 2-stage) in the offer inherently leads to the correction of both functions: **g(k)** and **gp(x,i)** for arguments **x** i **p** (decrementation by 1: $\mathbf{g(x) = g(x) - 1}$ and $\mathbf{gp(x,p) = 0}$).

(b) The criterion for the completion of the offer creating: $\mathbf{g(k) = 0, 1 \leq k \leq 4}$.

Graphically, the process of the density function correction is presented in Fig. 6.

The proposal of the construction of subdomains initiated density function (**PD**, **DP**, **OD**) is based on the connection of amount of information contributed by each domain with the prices of offered products:

$$\mathbf{gp(k,i)} = -\frac{\mathbf{pp_{k,i}}}{\mathbf{c_{k,i}}}\mathbf{log_2 pp_{k,i}}, \tag{5}$$

where

gp(k,i) – the initiated estimated value of the density function for the **i**-th object occurring in **k**-th domain,

$\mathbf{pp_{k,i}}$– the probability of occurrence of a specific object characteristics and (demand, interest, sales), resulting from the market research. It is therefore a fractional prevalence rate of the characteristics acquired as a criterion,

$\mathbf{c_{k,i}}$ **–** the price of the **i**-th object, registered in the **k**-th offer domain.

General payoff assessment (payoff - **PO**) of the player participating in the strategic game, identified with the thresholds configuration, determined in two stages will take the form:

$$\begin{aligned} &\mathbf{I}.\quad 0(\mathrm{x}) = \sum_{\mathbf{i=1}}^{\mathbf{lp}(\boldsymbol{x})} \left(-\frac{\mathbf{pp}_{\boldsymbol{x},\mathbf{i}}}{\mathbf{c}_{\boldsymbol{x},\mathbf{i}}}\mathbf{log_2 pp}_{\boldsymbol{x},\mathbf{i}}\right), \\ &\mathbf{II}.\quad \mathrm{PO(player)} = \sum_{x=1}^{4} \mathrm{Thd(player, x)} * 0(\mathrm{x}), \end{aligned} \tag{6}$$

lp(x) – the number of objects in the studied domain,

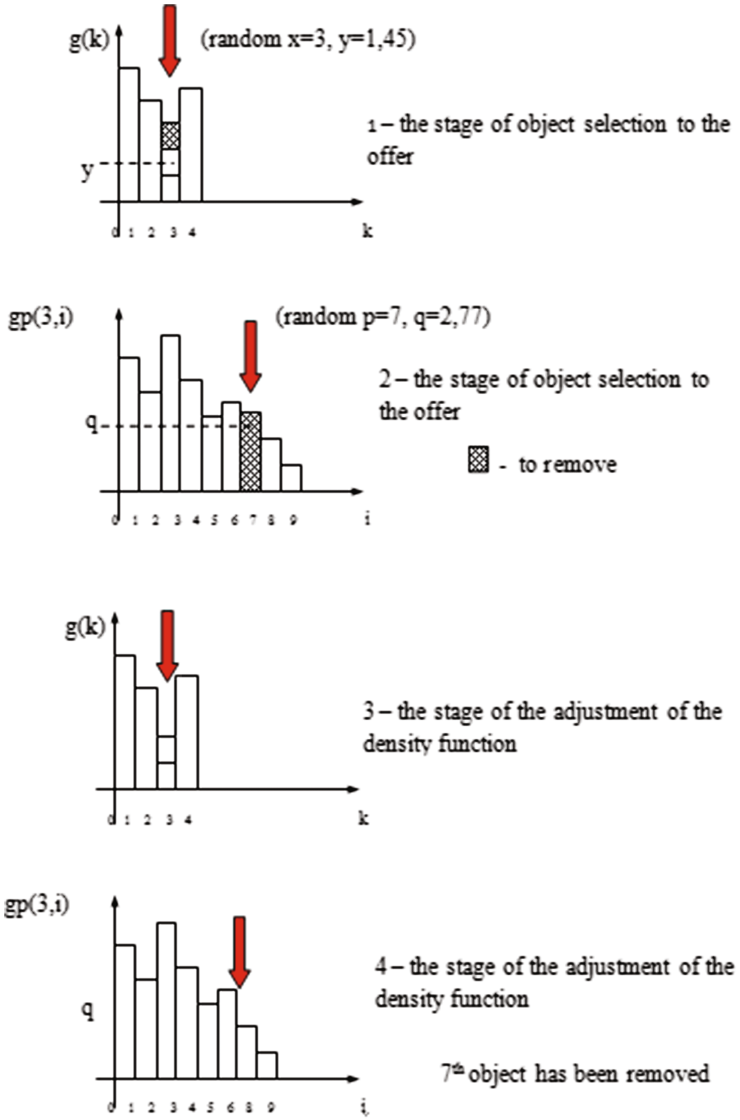

g(k)
(random x=3, y=1,45)
y
0 1 2 3 4
k
1 – the stage of object selection to the offer
gp(3,i)
(random p=7, q=2,77)
q
0 1 2 3 4 5 6 7 8 9
i
2 – the stage of object selection to the offer
▨ - to remove
g(k)
0 1 2 3 4
k
3 – the stage of the adjustment of the density function
gp(3,i)
q
0 1 2 3 4 5 6 8 9
i
4 – the stage of the adjustment of the density function
7th object has been removed

Fig. 6. Stages of the density function correction

Thd(player, x) – the configuration of thresholds represented by a given player (for all 4 domains).

It should be added that players' payoffs do not need to be standardized [5]. The game, described in the subsequent publications, has the form of a matrix and its effect are frequencies of use of offers represented by the players, expected values of payments, the game values, saddle points and other results typical for the strategic games [7].

4 Example of the Payoff Calculating for the Strategic Player

The typical data from market analysis relating to the volume of sales of products segregated by domains and standardized in their areas are shown in Fig. 7.

The data of products prices in the domains is located in Fig. 8.

Domain	ob. num.	pp(k,1)	pp(k,2)	pp(k,3)	pp(k,4)	pp(k,5)	pp(k,6)	pp(k,7)	pp(k,8)
D	7	0,052	0,167	0,077	0,21	0,016	0,087	0,37	
PD	3	0,42	0,256	0,32					
DP	8	0,049	0,32	0,058	0,069	0,26	0,19	0,012	0,041
OD	5	0,48	0,16	0,027	0,31	0,017			

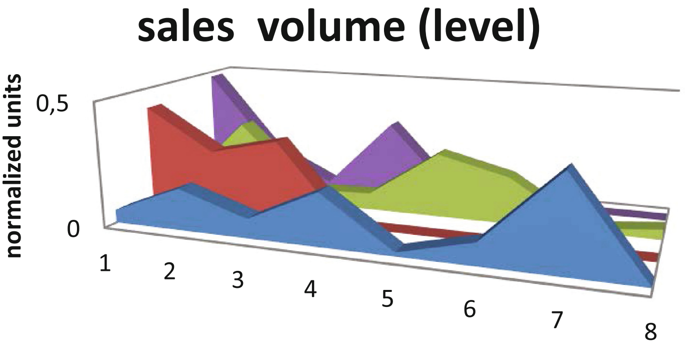


Fig. 7. Standardized levels of sales of objects in separate domains

Domain	ob. num.	c(k,1)	c(k,2)	c(k,3)	c(k,4)	c(k,5)	c(k,6)	c(k,7)	c(k,8)
D	7	178,20	112,80	16,80	80,50	104,99	190,10	139,60	
PD	3	82,00	25,10	148,60					
DP	8	77,20	74,00	44,00	198,60	126,80	90,90	80,70	128,50
OD	5	103,30	117,65	159,00	100,40	41,80			

Fig. 8. Prices of products in particular domains

Proposed, initiated density functions are calculated with the use of the phrase (5) and shown in Fig. 9.

Normalization accelerates the process of creating the offer; so it is worth using it. Finally, one comes to transformations (6) that is to the components estimation **O(x)** and level of the player's payoff **PO** (Fig. 10).

Domain	ob. num.	gp(k,1)	gp(k,2)	gp(k,3)	gp(k,4)	gp(k,5)	gp(k,6)	gp(k,7)	gp(k,8)
D	7	0,00124	0,00382	0,01695	0,00587	0,00091	0,00161	0,00380	
PD	3	0,00641	0,02005	0,00354					
DP	8	0,00276	0,00711	0,00541	0,00134	0,00398	0,00501	0,00095	0,00147
OD	5	0,00492	0,00360	0,00088	0,00522	0,00239			

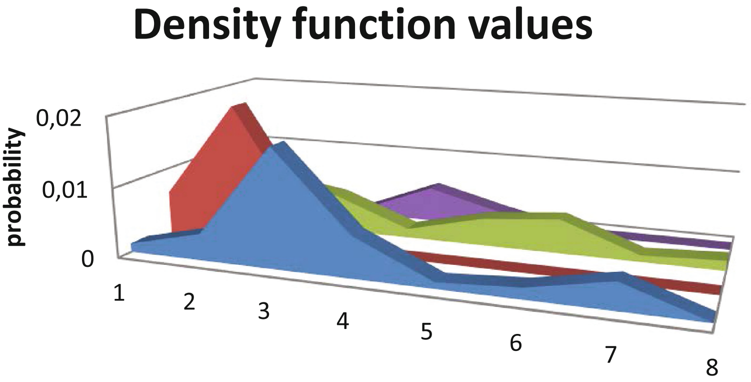


Fig. 9. The values of the density function (before normalization) for particular domains

Domain	O(x)	Thd(player,x)
D	0,0342	3
PD	0,0300	4
DP	0,0280	2
OD	0,0170	1

PO
0,296

Fig. 10. Intermediate results and the final ones relating to the payoff volume for the strategic player.

Structural proposals provide payoffs for different players in the matrix mixed games. The game can be carried out for the given configuration of offers between competing firms (shops), agencies of one company or even within a single unit offers.

5 Summary

The suggested way of building product-offering is an intelligent alternative of responding to sales parameters (characteristics). Reaching the customer thanks to analyzing sales volumes, price and the level of provided information about product (estimated with the size of the entropy) appears not to be uniquely original, however, it is remarkably fast and convenient. Further researches on offer effectiveness, that is use of payoffs for particular strategies, easily lead to create and play zero-sum matrix game [10]. Moreover, it allows to estimate frequency of using various strategies representing players as well as game value and find saddle-point. All parameters mentioned are used by offeror and facilitate customer's product searches. Multiplying the elimination level (up to 2 stages) in von Neumann's method ensures inclusion of the random strategies selection of objects in the offers, according to the given empirical distribution for a set of four domains.

Playing games is the next level of research that will be describe in the future articles.

References

1. Qin, Z., et al.: E-commerce Strategy, pp. 1–33. Springer (2014)
2. Roberts, M.L., Zahay, D.: Internet Marketing: Integrating Online and Offline Strategies, pp. 26–28. Cengage Learning (2012)
3. Ptak, A., Bajdor, P., Lis, T.: The use of social media in European union enterprises - comparative study. In: The International Academic Forum (IAFOR), The Asian Conference on the Social Sciences 2016 (ACSS 2016), Kobe, Japan, pp. 299–311, 9–12 June 2016
4. Skalen, P., Hackley, C.: Marketing-as-practice. Introduction to the special issue. Scand. J. Manag. **27**, 189–195 (2011)
5. Myerson, R.B.: Game Theory. Harvard University Press, Cambridge (2013)
6. von Neumann, J.: Various technique used in connection with random digits. Nat. Bur. Stand. Appl. Math. Ser. **12**, 36–38 (1951). 376–383
7. Curiel, I.: Cooperative Game Theory and Applications: Cooperative Games Arising from Combinatorial Optimization Problems, vol. 16. Springer Science & Business Media (2013)
8. Gray, R.M.: Entropy and Information Theory, pp. 61–93. Springer Science & Business Media (2011)
9. Morris, P.: Introduction to Game Theory. Springer Science & Business Media (2012)
10. Rapoport, A. (ed.) Game Theory as a Theory of Conflict Resolution, vol. 2. Springer Science & Business Media (2012)

Principles of Mobile Walking Robot Control in Scope of Technical Monitoring Tasks

Oleksandr Radomskyi(✉)

National Aerospace University KhAI,
17 Chkalova Street, Kharkiv 61070, Ukraine
o.radomskyi@gmail.com

Abstract. The article describes control synthesis for mobile walking robot, which has an interval mathematical model. We proposed proof of concept for mobile walking robot, carrying load up to sixty percents of robot's mass. Control system includes a computing unit, seven servos, a three-axis accelerometer and a moving camera operating in a visible optical range. Results of interval algorithms synthesis are shown for robot as an object of automatic control (OAC), with interval mathematical model. We defined the operational mode range of the system, its structure and interval parameters of mathematical model. It is shown that setting of interval values allows adequate modeling and description of processes in dynamic systems such as mobile walking robots. Main goal of this paper is to show possibility of classical Bode Diagram modification and application for the synthesis of interval control.

Keywords: Control algorithms · Mobile walking robot · Four legged robot · Interval model · Bode Diagram

1 Introduction

The modern mobile robots analysis shows, that a range of tasks exist in a limited environmental space during usage of flying robot is energetically inefficient or impossible at all. Also there are surroundings where wheeled robots cannot operate efficiently because of inappropriate underlying surface. Limited environmental space is a closed industrial space with production facility, which has a number of obstacles, such as machines, communications and walls, which robot cannot pass.

When you solve tasks of industrial facilities for technical monitoring [2] by means of the compact flying drones it is necessary to overcome navigation difficulties and real-time obstacle avoidance in dynamical environment. Control of compact aircraft [1] in a three-dimensional space with dynamic and fast-moving obstacles is not a trivial task, it depends on a concrete kind of environment with production facility. For every single limited space with facility it is necessary to specify control conditions and limitations, or install more complex control algorithm and faster control unit. Complexity of navigation task, that aircraft needs to solve, determines time while an aircraft is still in flight but is not performing useful operations consuming limited amount of fuel.

For wheeled robots [3] one of the most significant limitations is the frequent change of surface level, where robot moves. To overcome such kind of limitations additional

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_36

ramps are needed to be installed to allow robot to move over stairs, doorsteps and other obstacles of this type which has the different height. The proposed proof of concept for mobile walking robot demands less power resources in comparison with aircraft, and has a possibility to overcome obstacles with height greater than wheel radius for the same power wheeled robot.

Let us consider a walking robot as one of the general kinds in scope of control theory, so common methods and approaches can be applied for modeling and study for this kind of objects from control systems point of view.

The common control theory task is the regulator synthesis for a closed-loop control system, which has, at general level, an object, sensors and a regulator. Usually regulator synthesis is a non-trivial task, because every concrete system operates in a complex environment with a number of variable and non-linear parameters, so only the most general of them can be mathematically formalized and included into a mathematical model. Regulator synthesis deals with between actual and required system characteristics in time and frequency domains.

A number of methods exist for the regulator synthesis that can be efficiently applied for only a concrete class of control systems and has no advantages over other classes. Common methods are State Space [17, 18, 20], Parameters Localization [19] and Frequency Domain [18, 20].

The most interesting one for current study is the regulator synthesis by means of Bode Diagram (BD) [18], as it is a graph-analytic method. This allows to study the properties of the system effectively, and keep balance between different system constrains while changing the properties of the regulator. The method allows to satisfy two systems quality requirements – the transition time and stability margin for a single iteration of the regulator synthesis. But despite this advantage, BD method cannot be applied in a classical form for certain objects, because it is required to choose operating point of the system. The choice of the system operating point is the basis of BD for the regulator synthesis, that is not applicable for the systems described by means of interval values, because the operating point is also the interval value.

2 Overview of Existing Technologies

The assembling of walking robots and research in relation to walking machines started in the second half of the previous century because of the extreme computing power growth. Nowdays a wide range of industrial and military walking machines exist, also robotic production companies are ready to achieve mass market housekeeping robots in the nearest future, as it is shown in [4, 5].

Such designer of the world class robotic systems can be mentioned as the USA company DARPA Boston Dynamics, which started in early 70s at Massachusetts Institute of Technology (MIT) robotics research laboratory, it presented their military robot carrier Big Dog in 2005 [6], that handles up to 110 kilograms of load and is able to follow soldiers in the area of military actions. Also DARPA presented a range of four-legged robots such as Cheetah, modeling different stages of cheetah run and achieving speed up to nineteen miles per hour, that is the world record for legged robots. Another one DARPA's robot is the Small Dog, equipped with a bunch of sensors and designed to be taught in the process of motion to produce algorithms over raw surface using artificial intelligence. In 2010 the Department of Robotics in Moscow State Technical University n. a. N.E. Bauman (MSTU) created two-legged robot [7] for humanoid walking dynamic research [8]. In 2012 Toshiba introduced a legged robot that can overcome steps of stairs, uneven terrain and is able to avoid low-lying obstacles, they carry out the work at the Fukushima Daiichi nuclear power plant, where people cannot go. In 2016 Boston Dynamics introduced the four-legged robot Spot Mini - hydraulic powered mobile machine, that can freely navigate in living premises with stairs, can open doors, operate with conventional lightweight stuff such as cups, clothes or home appliances using an installed grabber. In February 2017 the Handle Robot was introduced which is a four-legged two-wheeled machine. Two legs are equipped with wheels so robot moves on two wheels and can carry load with two other legs, balancing by using the whole body, robot can jump, drive on stairs with damping all body oscillations, in such a way only wheels and lower parts of legs oscillate.

3 Robot Description

The object of the study of this work is a mobile walking robot, which has a computer vision system for navigation and monitoring tasks.

A subject of the study is the synthesis of interval control for an object of automatic control, which has an interval mathematical model. The following functional diagram of the system is regarded here:

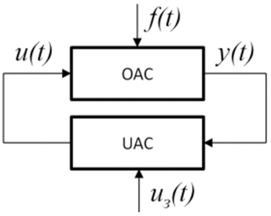


Fig. 1. Functional diagram of OAC

where: OAC – object of the automatic control, including sensors, a mechanical body and actuators; UAC – a unit of automatic control, it includes a regulator; *u(t)* – a control signal; *f(t)* – disturbance, forces and moments applied onto OAC while moving; $u_3(t)$ – setting signal; *y(t)* – output signal, the spatial position of the OAC.

The description of the OAC has one feature, expressed by, so called, the "center of mass oscillations" in relation to the motion trajectory, causing corresponding rotation and shift of principal axes. Such motion of object's principal axes is caused by specifics of chosen kinematic scheme.

In fact, an object has a set of non-linear parameters, including oscillation of center of mass, it makes it impractical to construct precise equations and includes all of them into a mathematical model. Otherwise it will be computationally inefficiently to use such mathematical model for modeling and study of dynamical object. So, mathematical model of proposed object in this work eliminates a set of nonlinear parameters by means of interval values, but preserves the main kinematic properties and motion specifications.

Mechanical design can be described as follows: a horizontal plate, where load is placed, mounded onto the main baulk in such a way that the weight of load distributes symmetrically along roll axis of robot. Bracings are placed under the main baulk accumulator batteries. Symmetrical front and rear moving baulks are connected to the front and rear sides of the main baulk via bearing joint, so they can freely rotate around corresponding vertical axes. Both front and rear moving baulks are equipped with three servos identically and symmetrically. On the moving baulk one servo is installed vertically for rotating as to the main baulk around vertical axis, two other servos are installed in orthogonal plane, equipped with pillars and allow robot to be located in vertical plane. So a pair of vertical servos provides rotating of moving baulks in horizontal plane while four other servos are used for moving pillars up and down. Camera is set on the compact crank mechanism which is triggered by a servo, so stereo image processing for navigation purposes can be achieved with one single camera. Advantage of such approach is saving computing power by retrieving one image at a time instead of two images at a time. Disadvantage of such moving camera unit is that camera cannot be calibrated so precisely as the fixed stereo-camera, but such method exists as [9, 10], proving that camera calibration is not mandatory.

3.1 Motivation

Redundant number of servo-drives require more power for keeping all kinematic links in the desired position. In the process of moving the walking machine only a few joints are used at full rate, the rest of joints, as a rule, don't move at all or move a bit.

Joints of living creatures has a complicated structure, comparing to robotic analogues, with number of kinematic links for achieving maximum utilization of energy for specific motion and high maneuverability of creature. The level of freedom for artificial walking machines is usually lower than that for living creatures. Forces and moments in joints of robot and in joints of creature are created by using fundamentally different means, so problem of energy optimization is a separate topic for discussion. Adding extra levels of freedom with additional kinematic link and corresponding muscles for limb of creature leads to better energy utilization during performing specific motion, but this cannot be applied to artificial limb. When creature's limb moves, a number of tissues are involved – muscles, tendons, fat, skin, bones and their joints, that provide damping, distribution of loads and are involved in forming of control forces and moments. For artificial limbs all of these functions, mainly, are performed by servo-drives attached to corresponding joint. So a number of tasks, that are performed in a creature's limb with no energy consumption because of tissues elasticity, in artificial limb are performed with usage of electricity for positioning servo-drives.

3.2 Distinction from Analogues

Based on assumptions which we have mentioned above, the proposed robot was designed so that it minimizes power consumption by excluding most of joints but still keep walking for transferring its mass. Future minimization of power consumption requires change of moving principle that is not in scope of this article.

The main distinction of the proposed robot kinematical scheme from the famous analogues with four legs is that the proposed scheme uses single rotational degree of freedom for lifting and putting down legs, while analogues are designed with at least two rotational degrees of freedom for every leg. That significantly increases power consumption for every movement that robot can perform.

Quality of walking, balancing and overcoming obstacles is lower in terms of robot orientation limits such as maximum body tilt and maximum obstacle height. But in terms of energy consumption the quality of walking is better because robot uses additional energy by increasing the length of pathway, only while dealing with obstacle avoidance, but not all the time, as analogues do, while moving on clean and straight surface.

3.3 Novelty and Benefits

A modification [11] of the classical Bode Diagram is used that allows application of BD for regulator synthesis when object's mathematical model includes set of interval parameters. It has been shown, that it is possible to apply classical Bode Diagram with modification for control system regulator synthesis, when an object of the automatic control has an interval mathematical model. The obtained regulator satisfies general requirements, such as stability margins and transition process time.

3.4 Mathematical Model

Mathematical model is non-linear and requires linearization for the correct regulator synthesis. Mathematical calculations are out of this paper scope, so only results will be shown.

In compact form, taking into account the initial state values and assumption, that resistant moments are static and applied only while moving, mathematical model appears as:

$$a_{11}\ddot{\varphi}_1 = b_{11}M_{kp} - sign(\dot{\varphi}_1)(b_{11}M_{c_4} + M_{c_{23}}) \tag{1}$$

$$a_{21}\ddot{\Psi}_1 + a_{24}\varphi_1 = b_{21}M_{kp} \tag{2}$$

$$v^e_{цм_x} = -a_{33}\dot{\varphi}_1;\ v^r_{цм_x} = -a_{43}\dot{\Psi} \tag{3}$$

$$v^r_{цм_y} = -a_{53}\dot{\Psi}_1;\ v^r_{цм_z} = -a_{64}\dot{\Psi} \tag{4}$$

where: φ_I – rotation angle of leading bridge AC (Fig. 2a); ψ – tilt rotation angle of robot (Fig. 2b); $v^e_{цм_x}$ center of mass absolute velocity vector; $v^r_{цм_x}$ $v^r_{цм_y}$ $v^r_{цм_z}$ –

projections of center of mass relative velocity onto XYZ principal axes (Fig. 2); $M_{кр}$ – servo-drive torque; M_{c4} – resistance at the end point of front robot foot, touching ground.

Output signal of object is the center of mass linear motion along X axis, that can be defined via integration of (3), (4):

$$x^{e}_{\text{цм}} = -a_{33}\,\varphi_1;\ v^{r}_{\text{цм}} = -a_{43}\Psi \tag{5}$$

$$x_{\text{цм}} = -a_{33}\,\varphi_1 - a_{43}\,\Psi \tag{6}$$

Robot motion and applied forces and moments are shown below:

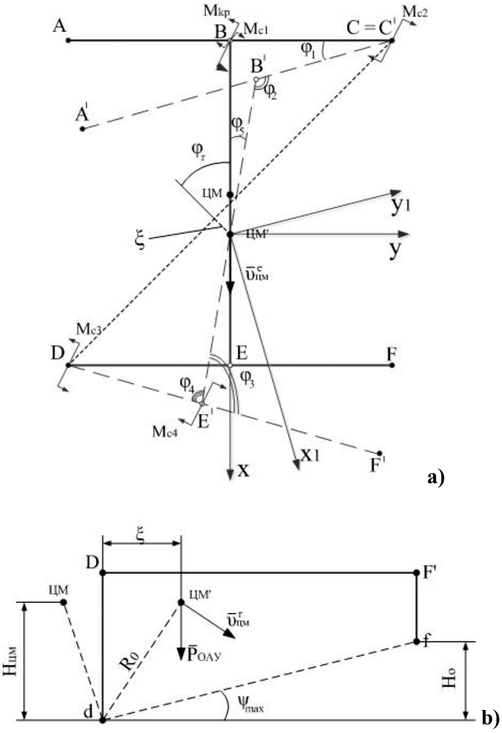


Fig. 2. Motion in (a) horizontal XY plane (b) vertical plane of robot tilt while walking

where *A*, *B*, *B'* , *C*, *C'*, *D*, *E*, *E'*, *F*, *F'* – are border points of constant length moving parts, M_i – are moments around vertical axis, φ_i – are variable angles between parts while moving, *ЦМ* and *ЦМ '* – is a center of mass at initial position and position after robot performs single step respectively, ξ – is a center of mass shift, R_0 – is a tilt radius, $\overline{v}^{e}_{\text{цм}}$ and $\overline{v}^{r}_{\text{цм}}$ – center of mass interval velocities relative to Y axis and R_0 respectively, $\Psi_{\max}$ – maximum tilt angle, $H_{ЦМ}$ – center of mass height, H_0 – height of robot leg lift, $\overline{\text{P}}_{\text{OAY}}$ – robot's force of gravity as $m_{robot}g$, f and d are projections of F and D onto surface where robot moves.

Object is a four legged robot, that performs motion step-by-step by keeping two diagonal legs (Fig. 1 *C* and *D*) on the surface and moving two other diagonal legs (Fig. 1 *A* and *F*). On above figure one step is shown, so legs *C* and *D* are static, legs *A* and *F* performs transition to A' and F' respectively, while moving, robot's center of mass *ЦМ* also moves to the new position *ЦМ* $'$.

3.4.1 Discrete Model

For numerical modeling and calculations it is necessary to acquire discrete model. Discrete model can be acquired from continuous equations via time discretion with Euler scheme:

$$\frac{dx}{dt} \approx \frac{x([k+1]T_0) - x(kT_0)}{T_0} \tag{7}$$

where T_0 – quantization period. Value of T_0 ranges as $T_0 \in [T_{0min}, T_{0max}]$, where bottom border is defined by discrete process stability criteria, and upper one – from quality of continuous signal based on its discrete series (Kotelnikov–Shannon theorem) [12].

To find T_{0max} for upper and bottom borders interval BD it is possible to write object's transfer function performing Laplace transform for zero initial condition:

$$W(s) = \frac{X(s)}{U(s)} = \frac{\bar{K}(\bar{T}_1^2 s^2 + 2\bar{\varsigma}\bar{T}_1 s + 1)}{s^3(\bar{T}_2 s + 1)} \tag{8}$$

where U(s) – s-domain control signal, X(s) – s-domain linear motion of OAC center of mass, $\bar{K}$ – interval gain coefficient of OAC, $\bar{T}_1$ $\bar{T}_2$ – interval time constants, $\bar{\varsigma}$ – interval damping coefficient. For computation of constants from (8) using relation for center of mass motion (6) and relations (1), (2):

$$\bar{K} = -\frac{a_{43}a_{24}a_{11}b_{11}}{a_{21}} M_{kp} \tag{9}$$

$$\bar{T}_1 = \sqrt{\frac{a_{33}a_{21}b_{11} + a_{12}a_{43}a_{11}^2 b_{21}}{a_{11}a_{12}a_{24}a_{43}b_{11}}} \tag{10}$$

$$\bar{T}_2 = \frac{a_{11}}{a_{12}}; \bar{\varsigma} = \frac{a_{12}b_{21}}{a_{24}b_{11}} \cdot \frac{1}{2\bar{T}_1} \tag{11}$$

where $\bar{K} = [-63.05; -0.72]$, $\bar{T}_1 = [0.94; 16.45]$, $\bar{T}_2 = [28.83; 32.12]$,

$\bar{\varsigma} = [3.98 \cdot 10^{-8}; 3.24 \cdot 10^{-5}]$. Next, necessary to perform transition into frequency domain with $s = j\omega$. Equation of robot Actual Bode Diagram (ABD) (Fig. 6 L_{Amax} and L_{Amin}):

$$L_p(\omega) = 20lgK - 60lg\omega - 20lg\sqrt{T_1^2\omega^2 + 1} + 20lg\sqrt{(1 - T_1^2\omega^2)^2 + (2\varsigma T_1\omega)^2} \quad (12)$$

3.4.2 Synthesis of Regulator

The initial step for correct interval characteristics computation is choosing system's operational point. Numerical modeling was performed, and result level-lines in plain of regulator parameters was plotted (Figs. 3 and 4). Plot of level-lines shows stability margins and cutoff frequencies for both "maximal" L_{Amax} and "minimal" L_{Amin} borders (Fig. 6) of interval ABD.

Regulator is the first order integrating-differential unit (13), stability margins will be computed in plain of regulator parameters. Regulator transfer function:

$$W_{REG}(s) = \frac{v(s)}{\varepsilon(s)} = K_{REG}\frac{T_{1REG}s + 1}{T_{2REG}s + 1} \quad (13)$$

where W_{REG} (s) – is the regulator transfer function, K_{REG} – regulator gain, T_{1REG} ϵ {0,0.1,…,3.0} – is a first time constant of regulator and T_{2REG} ϵ {0,0.01,…,2.0} – is the second.

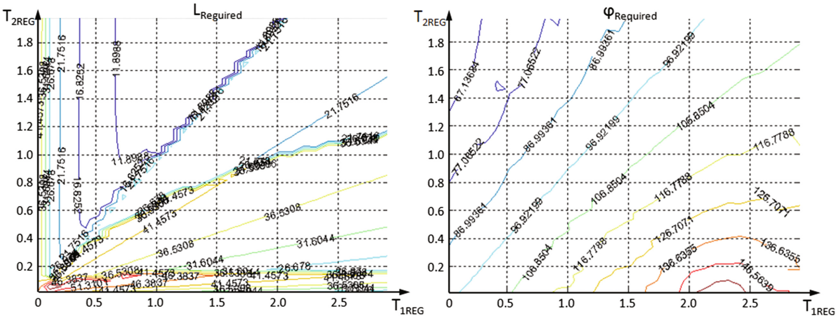


Fig. 3. Plain T_{1REG} T_{2REG} of regulator parameters for maximal Bode Diagram border L_{Amax} amplitude $L_{Required}$ and phase $\varphi_{Required}$ stability margins

From a number of available stability margins values (Figs. 3 and 4) three points T_{REGi} were chosen and transitions processes are plotted (Fig. 5) for maximal and minimal borders of ABD L_{Amax} and L_{Amin} (Fig. 5) respectively T_{REG01} = [0.3;0.08], T_{REG02} = [0.75;0.15] and T_{REG03} = [1.5;0.2].

Characteristics Fig. 5 confirm that trying to make system more stable causes the worst system response to control commands and transition process time increases. So stability margins were chosen respectively to minimal available transition process time.

Transfer function of closed-loop system:

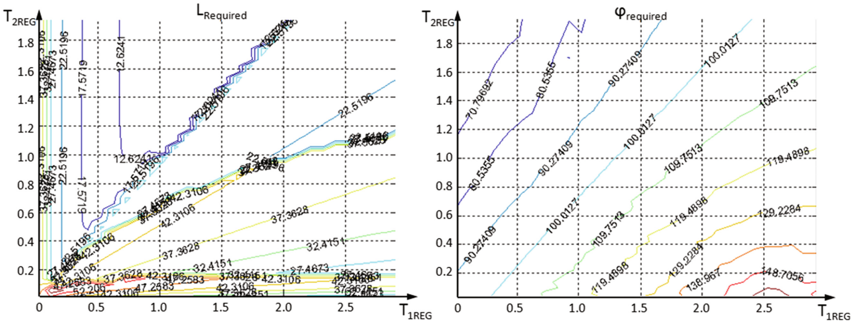


Fig. 4. Plain T_{1REG} T_{2REG} of regulator parameters for minimal Bode Diagram border L_{Amin} amplitude $L_{Required}$ and phase $\varphi_{Required}$ stability margins

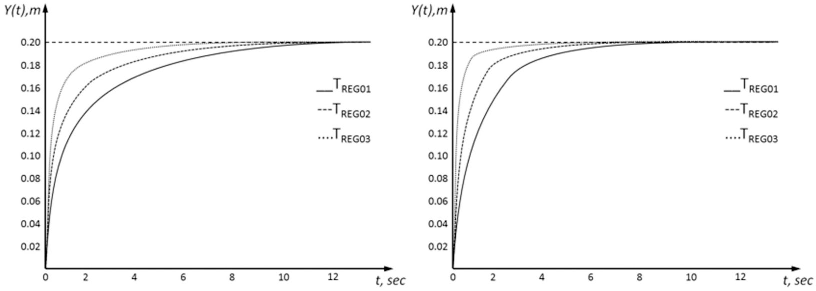


Fig. 5. Transition processes for maximal(left) and minimal(right) borders of Bode Diagram with different parameters of regulator T_{REGi}, where *Y(t),m* is center of mass shift

$$\Phi(s) = \frac{E(s)}{U(s)} = \frac{W_{REG}(s)W(s)}{1 + W_{REG}(s)W(s)K_s} = \frac{11.69s^3 + 38.97s^2 + 0.043s + 0.14}{2.3s^5 + 28.91s^4 + s^3} \tag{14}$$

where: E(s) – s-domain error signal, difference between required and actual object's center of mass linear transition; U(s) – s-domain control signal; K_s = 5 – distance sensor's gain.

Stability margin is chosen according to general requirements for control systems, overshoot *δ = 20%*. Transition process time t_{tp} *= 25* s is chosen depending on system technical characteristics and fixed moving distance from start to finish *S = 2* m. Proper values of stability margin and corresponding transition process time are required for obtaining cutoff frequency of classical Bode Diagram. Cutoff frequency (15) is chosen by means of Diagram of Solodovnikov [13].

$$\omega_{cf} = \frac{3.2\pi}{t_{tp}} = 0.4 \tag{15}$$

Such cutoff frequency causes, that low-frequency part of ABD will conjugate with low-frequency part of Required Bode Diagram (RBD) (Fig. 6 L_{Rmax} and L_{Rmin}) at frequency about 10^{-4} rad/sec that is physically impossible to achieve with the given robot. One of the possible solutions is a higher tilt of regulator BD (Fig. 6 L_{REGmax} and L_{REGmin}) low-frequency part, but it causes higher order of magnitude for regulator transfer function. Another solution is increasing of regulator gain, that causes shift of RBD up along vertical L axis (Fig. 6). When increasing regulator gain conjugate frequency on low-frequency parts are shifts to the right, decreasing order respectively.

After a few iterations of RBD plotting with different shifts of regulator BD characteristic, upward shift by +30 dB was chosen. Such value of upward shift provides required values of system quality, regulator gain that is possible to achieve an doesn't require increasing of regulator transfer function order, that may be caused by tilting regulator low-frequency BD part (Fig. 6).

Additionally, there is no need to perform static analysis of the system because denominator of transfer function (8) has third order of complex number s^3 that causes zero error with step, linear and square input signals.

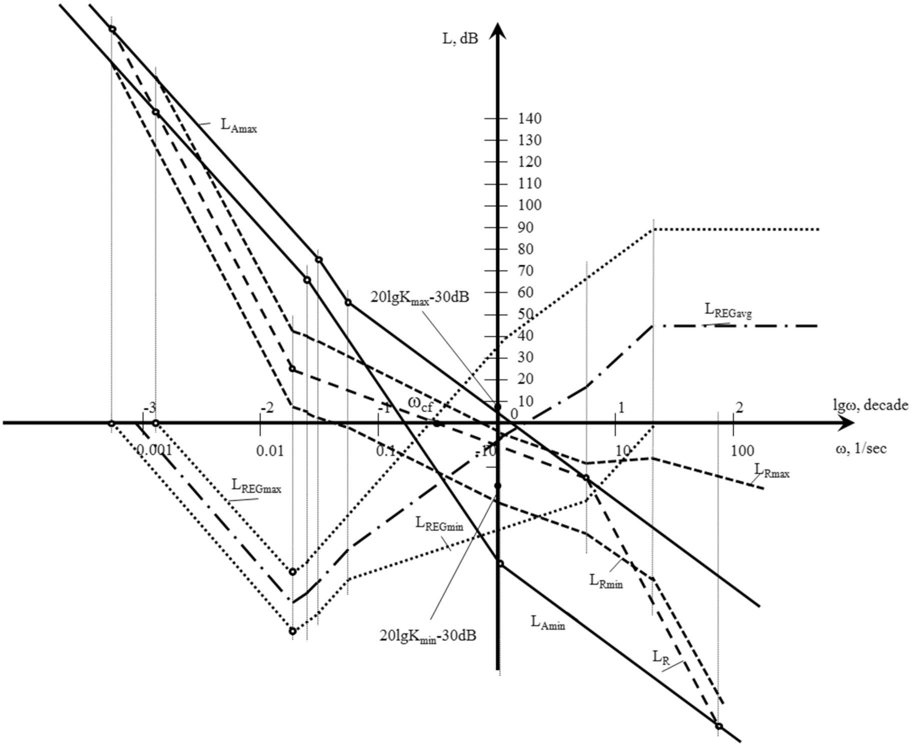


Fig. 6. Interval Bode Diagram of the system

3.4.3 Modeling of Discrete System

To ensure system stability after discretion, modeling of digital regulator has been made with the aim of determining of quantization period T_0. Analysis of closed-loop system (14) stability performed with trajectories of system's zero-poles.

With chosen $T_{REG01} = [0.3;0.08]$ model of discrete closed-loop system has been formed by means of Matlab Simulink (Fig. 7) the transition process for step input signal has been acquired (Fig. 8).

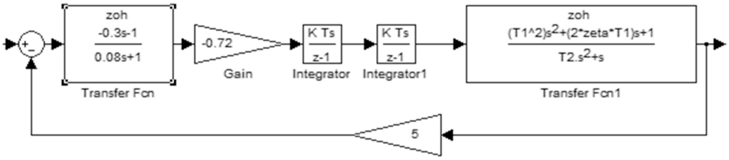


Fig. 7. Discrete closed-loop system Simulink model

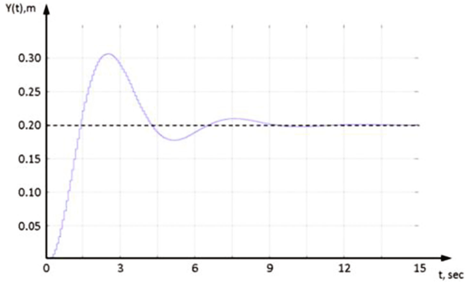


Fig. 8. Discrete closed-loop system response for a step input signal

Using $T_{REG01} = [0.3;0.08]$, the numerical modeling has been performed with T_0 ranging from 10^{-3} to 0.5 with single increment of 10^{-3}. Zero-poles trajectories shows that discrete system is stable at whole range, so quantization period T_0 was chosen equal to 10^{-3} s.

3.5 Robot Tasks and Control

The sphere of robot's usage can be limited to a few groups such as tasks where continuous monitoring without human being and the preliminary monitoring of damaged or dangerous facilities are needed.

A control principle for such kind of object can be downscaled to the control of servo-drives. Current version is equipped with TowerPro MG996 servo drives, controlled by power width modulation. Working range of this servos from is 0 to 180° and depends on control pulse width on 50 Hz frequency.

Control system consists of the main unit, which is responsible for servo-drives control following operator commands, or commands acquired from higher level control

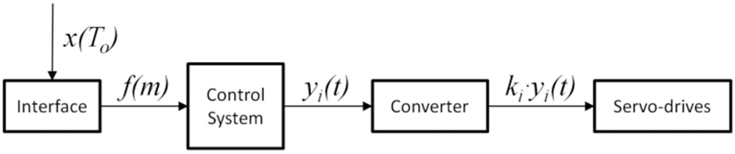


Fig. 9. System structure

loop and regulator unit, that is responsible for adaptation of control commands respectively to sensors data. While waiting for new command, system continuously repeats the last command. So when operator or higher control loop are preparing control command that determines type of movement, the main unit computes relations of pulses width changes in time. Next, regulator unit uses real-time data from sensors and updates computed pulses width so robot can keep on moving with no loss of efficiency (Figs. 9 and 10).

Where: *x*(T_0) input signal in binary form, *f*(*m*) control command, $y_i(t)$ servo-drive actual control signals, k_i gain for *i*-th servo-drive actual control signal, $i \in \{1, 2, \ldots, 7\}$.

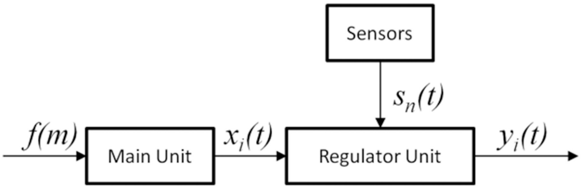


Fig. 10. General schematic of control system

Where: $x_i(t)$ is a servo-drive general control signals, generated from default set of robot movements, $s_n(t)$ data from *n*-th sensor, $n \in \{1, 2, 3\}$.

Robot control algorithms computed using principle, described above - first, control command *x*(T_0), that passed to the system input Interface. Next, after receiving command, values for servo-drives control, delays and control command *f*(*m*) are computed. Control system computes relations for servos $x_i(t)$ using default preset of robot motions, and then retrieves data from sensors. Using sensors data Regulator Unit updates values for servos control (Fig. 11).

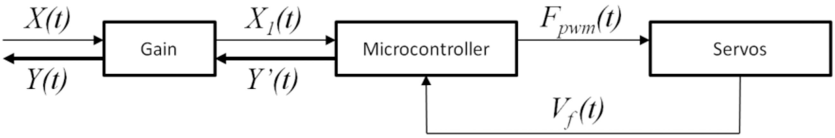


Fig. 11. General structure of control unit

Where: *X(t)* general control signals, $X_1(t)$ control signals for microcontroller, $Y'(t)$ controller output signals, $F_{pwm}(t)$ control pulses for servo-derives, $F_f(t)$ feedback voltage (Fig. 12).

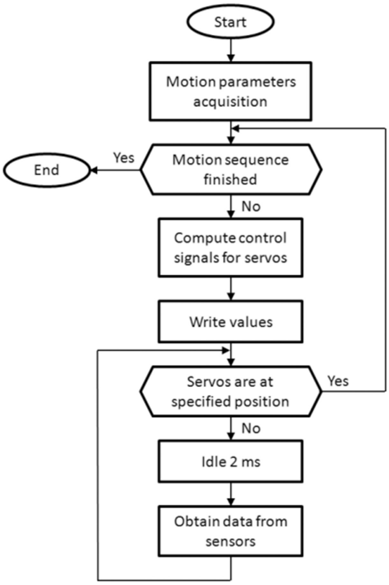

Start
Motion parameters acquisition
Yes
End
Motion sequence finished
No
Compute control signals for servos
Write values
Servos are at specified position
Yes
No
Idle 2 ms
Obtain data from sensors

Fig. 12. Generalized control algorithm

When robot loses its dynamical stability while moving, say flip around roll axis because of some sort of disturbance, design of robot body parts allows to recover to operational condition, it is meant the normal position of robot relative to the ground, by performing of a set of specific movements. First, when robot loses dynamical stability, control system performs actions to restore statically the stable state of robot body. Then, robot performs a set of motions to change relative position as to the center of mass point in such a way that the body rotates around roll axis at 180°. Also, while restoring operational condition, robot body rotates at about 30° around yaw axis, so additional movements are required to turn back to the robots route.

4 Conclusion

Shown control principles and general schematic of control system for mobile robot above can be used for researching different operational conditions of both walking robots and the design similar to that which has been mentioned above.

Proposed system uses only units which are involved in regular motion for restoring normal orientation when robot loses stability and falls down. Pairs of images for stereo reconstruction of observed surrounding obtained with only one camera, which is set on the compact crank mechanism, that twice minimizes the cost of the hardware required for stereo image acquisition.

It has been shown, that classical BD can be modified and applied for dynamic objects having mathematical model with interval values. The results of mobile walking robot non-linear mathematical model linearization and the discrete model with all required checking of result system stability were provided. The results for stability range computation in plain of regulator parameters are shown and transition processes for some chosen points are modeled. The range of operating point is chosen. The closed-loop transfer function is acquired and Interval Bode Diagram (IBD) is plotted according to the system requirements such as stability margins and transition process time.

References

1. Matveyev, S.I.: Vysokotochnyye sistemy RV i A: perspektivy i osnovnyye napravleniya rabot po sozdaniyu razvedyvatel’no-udarnykh i razvedyvatel’no-ognevykh kompleksov. In: Matveyev, S.I. (eds.) [tekst] Voyennaya mysl’, vol. 2, pp. 23–27 (2003)
2. Myasnikov, Y.V.: Vysokotochnoye oruzhiye i strategicheskiy balans (izdaniye Tsentra po izucheniyu problem razoruzheniya, energetiki i ekologii pri MFTI). (Tekst)/ Ye.V. Myasnikov – Dolgo-prudnyy, p. 43 (2000)
3. Dergachov, K.YU., Flerko, S.M., Kravtsov, D.V.: Metodika viznachennya optimal’nikh marshrutív rukhu rukhomikh ob’êktív u kompleksí zadach komandnogo punktu dispetchers’koï sistemi [Tekst]. In: Yu, K., Dergachov, S.M., Flerko, D.V. (eds.) Kravtsov Sistemi obrobki ínformatsíï. NANU, PANM, KHVU, Kharkív. vol. 1, pp. 213–217 (2005)
4. Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., Abbeel, P.: Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In: Proceedings of IEEE International Conference on Robotics and Automation (2010)
5. Saxena, A., Driemeyer, J., Ng, A.Y.: Robotic grasping of novel objects using vision. Int. J. Rob. Res. **27**(2), 157–173 (2008)
6. Buehler, M., Playter, R., Raibert, M.: Robots step outside. In: International Symposium on Adaptive Motion of Animals and Machines (AMAM), Ilmenau, Germany, pp. 1–4, September 2005
7. Koval’chuk, A.K., Kulakov, D.B., Semenov, S.Y., Lomakin, V.O.: Sintez funktsiy upravleniya khod’boy dvunogogo shagayushchego robota pri pomoshchi uproshchennoy matematicheskoy modeli. Nauka i obrazovaniye: Elektronnoye nauchno-tekhnicheskoye izdaniye (2010)
8. Loffler, K., Gienger, M., Pfeiffer, F.: Sensor and control design of a dynamically stable biped robot. In: IEEE International Conference Robotics and Automation, Proceeding. ICRA APOS 2003. W.t. 2003. vol. 1(14–19), pp. 484–490 (2003)
9. Bab-Hadiashar, A., Suter, D.: Robust optical flow computation. Int. J. Comput. Vis. **29**, 59–77 (1998)
10. Bruhn, A., Weickert, J., Schnorr, C.: Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. Int. J. Comput. Vis. **61**(3), 211–231 (2005)
11. Modeli i metody sinteza sistemy avtomaticheskogo pozitsionirovaniya rezhimov vikhrevogo energorazdelitelya (Tekst): dis. … kand. tekhn. nauk: 05.13.03/Pasichnik Sergey Nikolayevich; Nats. aerokosm. un-t im. N. Ye. Zhukovskogo “Khar’k. aviats. in-t”. - KH., vol. 158(l), ris. - Bíblíogr.: ark, pp. 146–158 (2011)

12. Tekhnicheskaya kibernetika. Teoriya avtomaticheskogo regulirovaniya. Kniga 2. Analiz i sintez lineynykh nepreryvnykh i diskretnykh sistem avtomaticheskogo regulirovaniya [Tekst]. In: Solodovnikova, V.V. (ed). Pod red, Mashinostroyeniye, p. 682 (1967)
13. Solodovnikova, V.V., Mashgiz, M.: Osnovy avtomaticheskogo regulirovaniya. Pod red (1954)
14. Kulik, A.S.: Identifikatsiya matematicheskoy modeli vikhrovogo energorazdelitelya v chastotnoy oblasti (Tekst). In: Kulik, A.S., Pasichnik, S.N. (eds.) Aviatsionno-kosmicheskaya tekhnika i tekhnologiya, vol. 7(94), pp. 192–196 (2012)
15. Tsybakov, B.S., Iakovlev, V.P.: On the accuracy of restoring a function with a finite number of terms of Kotel'nikov series. Radio Eng. Electron. **4**(3), 274–275 (1959)
16. Dugarova, I.V.: Application of interval analysis for the design of the control systems with uncertain parameters, Ph.D. thesis, Tomsk State University, Russia (1989)
17. Smagina, Y., Brewer, I.V.: Robust modal P and PI regulator synthesis for plant with interval parameters in the state space. In: Proceedings of ACC, Chicago, Illinois, USA, pp. 1317–1321 (2000)
18. Dorf, R.C., Bishop, R.H.: Modern Control Systems, 11th edn. Prentice Hall, Upper Saddle River (2008)
19. Vostrikov, A.S.: Sintez sistem regulirovaniya metodom lokalizatsii, p. 251. NGTU, Novosibirsk (2007)
20. Dorf, R.C., Bishop, R.H.: Modern Control Systems. Addison-Wesley company, New York (1990)

Computer Systems – Simple, Complicated or Complex

Dominik Strzałka(✉)

Faculty of Electrical and Computer Engineering,
Rzeszów University of Technology, Al. Powstańców Warszawy 12,
35-959 Rzeszów, Poland
strzalka@prz.edu.pl

Abstract. The notion system, in a wide range of disciplines from ecology to physics, social sciences and informatics, has received significant attention in the last years. The behavior of each system can be understood when the proper approach is taken. In the case of computer systems there is also a need to have a paradigm change in approach to their analysis (and synthesis) because they are complex systems. The main goal of this paper is to present a few examples (reasons) that will justify, why the computer systems are the complex systems and why the complex systems approach should be taken.

Keywords: Complex systems · Computer systems · Long-range and long-term dependencies

1 Introduction

Generally, the notion *system* can be interpreted as a structure consisting of the large number of interdependent elements. Notwithstanding, in order to understand the behavior of any system one must understand not only the behavior of its individual elements, but also how they act together [1]. The simple system approach assumes that the whole knowledge about the system is possible when the behavior of each system component is known, but this approach fails when the complex systems are analyzed – in such a case one needs the complex systems approach, which should be also used in the case of nowadays computer systems. The justification for this statement cannot be given by one, simple example or by one reason, because it is rather a paradigm change in approach than a simple evolutionary process [2]. However, this paper presents a discussion which should help understanding why the nowadays computer systems evolve towards complex systems governed by dependencies that are long-range in time and space domain.

There are two types of systems: simple and complex. The simple systems consist of relatively small number of elements that act together according to well-defined and well understandable laws. A pendulum can be the example of such a system, its behavior is well described in a completely deterministic manner. In the case of complex systems there is not one commonly acceptable definition, but it seems that complex systems can be described as systems that usually are built of many, however (it is not a rule) not necessary, identical elements cooperating together according to the rules, which are not

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_37

well defined, but also can change with time. Moreover, the character of dependencies between the system component parts can be very plastic and also can undergo not necessary defined changes in time. A flock of flying ducks can be the example of such a system. Sometimes one can find the third type of systems – complicated ones, but this is only a kind of simple systems. The complicated systems are built from a huge number of elements, but with a function well-defined to realize, governed by understandable laws similarly to the simple systems [3]. Such systems fit well to the conception of Deutsh clock mechanism [4]. The best example of such a system is the steam locomotive, which is also the classical example of machine. The traditional and broadly accepted definition of the machine is related to physics. It assumes that the machine is considered as a physical system working deterministically in basic well-defined, physical cycles, built by man, and intended to concentrate the dispersion of the energy in order to do some physical work [5]. Thus, steam machine works almost[1] according to Deutsh conception [4] as a perfect mechanism that moves cyclically according to the well-known and described laws of physics; it is a complicated but simple system.

The whole paper is organized as follows: after preliminaries, in Sect. 2 we have a discussion about Turing machines and a change in their perception. Section 3 shows some examples that justified why computer systems evolved towards complex systems. In Sect. 4 an example of statistical mechanical analysis of one sorting algorithm is given in order to make a connection between Turing machines and the concept of energy transformation that is done during processing. Conclusions are presented in Sect. 5.

2 Turing Machines and Their Limitations

In the case of computer systems there is also a very important (fundamental) definition of machine – the Turing machine. This is a brilliant idea of Alan Turing [6] who was searching the solution for some cases of Hilbert's *Entscheidungsproblem* [7], i.e., he was looking for a mathematical model that will be able to solve (after the discovery of Gödel [8]) not necessary all computable problems, but some of them that are well-defined. Turing machine is a mathematical, but not a physical model. It consists of simple data structure mostly identified with a sequence of symbols or a tape, head and program. The tape has got infinite length that in spite of expectations is not without meaning, because the existence of tape limit allows only possibility of having finite number of machine configurations thus Turning machine will be reduced to the finite automaton [9]. The accessible operations allow for moving cursor left and right, write on a current cursor position and to decide about the further direction of calculations according to the value of current symbol. This model works in a deterministic manner. One can read this in Turing paper where he wrote [6]: "*The possible behavior of the machine at any moment is determined by the m-configuration* [of conditions] q_n *and the scanned symbol* $S(r)$. *This pair* q_n, $S(r)$ *will be called the "configuration": thus the*

[1] Description *almost* is due to the fact that Deutsh mechanism can be moved forward and backward while from thermodynamics it's known that each system should undergo *the arrow of time* rule.

configuration determines the possible behavior of the machine". This model represents a simple, deterministic system. Moreover from Alonso Church thesis it is known that [10]: "*Each reasonable attempt to create mathematical model of algorithmic computing and definition of its working must lead to the model of computing and connected with it measure of time cost, which are polynomial equivalent to Turing machines*" so Turing machine is a model of computing that is a basis of theoretical computer engineering – engineering based on a simple systems paradigm. But it should be noticed that the physical implementations of Turing machines have got a few important properties that allow for modeling algorithmic computing. Among them there are [9, 11]:

- their computations are *closed*, i.e., the influence of surroundings is excluded which also means that the values of each input should be known prior before the computing begins;
- their resources (time and memory) are limited, but in a mathematical model the length of tape used during computations is infinite;
- their behavior is strictly determined, which results from the fact that computations always begin for the same starting configuration and do not depend on time.

The implementations of Turing machine can be built in many different ways (one can imagine Turing machine as a steam machine) probably with many elements, but because assumptions given by Turing assume that they work in a deterministic way such implementations will be at least complicated systems (i.e., still simple systems). But the whole scientific community of computer engineers should realize that nowadays computer systems also compute in mode that is definitely different one than algorithmic because computers [9, 12]:

- usually work in an interactive mode (it is a mode that is different from algorithmic processing);
- exist in the environment with almost unlimited resources of memory (it should be noticed that the existence of the Internet can be treated like a physical implementation of unlimited resource of memory);
- work continuously (nowadays computers and managing them operating systems work continuously even for several tens that can put the halting problem in question);
- process unfinished streams of dynamically generated queries on input (state of output of such a system is defined by both: actual state of input and history of executed by system computations);
- evolve by the changes of its physical structure (in many computer systems there is a possibility to plug in, disconnect and on-line configuring different types of devices during a normal system work).

A mode with these properties is called the interactive one and it can put in a question all so far done achievements of theoretical informatics. Thus it seems also that the existence of interactive mode of processing is the first important reason that should change our view on computer systems, because they are complex ones. For the first time this opinion was expressed by Dijkstra in 1972 (details are in [11] where we can read (Sect. 5): "*It was pointed out by Dijkstra that the structural complexity of a large software system is greater than that of any other system constructed by man*"). Similar remark was given by M. Gell-Man – in [39] we can read: "(…) *chose topics that could be helped along by these huge, big, rapid computers that people were talking about – not only because we can use the machines for modelling, but also because these machines themselves were examples of complex systems*".

The above deliberations seem to have only *academic form*, but as an example the definition of notion algorithm can be showed. Knuth wrote in [13]: "*an algorithm has five important features: finiteness* (…), *definiteness* (…), *input* (…), *output* (…), *effectiveness* (…)". All of these features are shortly explained. But the description of input in the first and second edition of his famous book is: "*An algorithm has zero or more inputs, i.e., quantities that are given to it initially before the algorithm begins. These inputs are taken from specified sets of objects* (…)", while in the third edition one can find: "*An algorithm has zero or more inputs, i.e., quantities that are given to it initially before the algorithm begins or dynamically as the algorithm runs. These inputs are taken from specified sets of objects* (…)". So in the third edition Knuth has changed this point and allowed for the possibility of interaction during algorithmic processing. Between algorithms and Turing machines there is an equality (Papadimitriou wrote that [14]: "*One can think of Turing machines as the algorithms*"), so the data input during computations should be excluded. Dina Goldin, a co-author of [9, 12] in private correspondence wrote that she believed that Knuth did not change the definition allowing dynamic data input, however, she also confirmed that in their private talk Knuth told her that he considered such a possibility. As one can see the definition has been changed.

3 Complex Computer Systems

As it was mentioned above the interactive processing is a kind of *bridge* that connects computer systems with the complex systems because the problem of interactions (the influence of surroundings) is taken into account. Because the nowadays processing is at least both algorithmic and interactive there is a real need to describe the influence of task, which will be processed, on a computer system. A computer processing should be viewed holistic so the task, the algorithm and the executing structure should be analyzed together. This is a different approach than is now commonly used in the case of algorithmic computing, because from classical theory of algorithmic complexity it is known that the problem of algorithm complexity is analyzed with minimization of the influence of specific task instances [15] and (usually) described only by the big O notation. In the above proposed holistic analysis a real behavior of algorithm is analyzed together with the dependences that exists in task; the classical approach is changed [16]. The interactive and algorithmic processing coexist in computer systems so their behavior will be also determined by properties of task (input data). If input data (task, batch) will have the non-correlated structure[2] in time and space it won't influence

[2] It is hard to shortly describe this statement, because there can be a different types of dependencies (time correlations – for example the values for succesive elements that should be sorted can udergo some time statistical properties; spatial correlations – for example the analyzed structure of graph can be nor regular nor random; etc.) in task structure.

the processes described by deterministic algorithm, i.e., the input structure will be neutral[3]. This neutrality means that input data won't for example lead to the existence of long-range dependencies in processing dynamics [16].

3.1 Long-Range Dependencies – Time Domain

What are the long-range dependencies? This problem was presented for the first time in the fifties of the XXth century when H. E. Hurst, thanks to the R/S analysis, showed that sets of measurements of the discharge of the Nile, other rivers and some natural phenomena are governed by dependencies that are long in time domain [17]. Two interesting terms: the *Joseph effect* and the *Noah effect* were introduced to show that, according to famous paper of B. B. Mandelbrot [18], time series can be statistically self-similar. This discovery was very important because it showed that many natural phenomena (physical systems) are governed by dependencies that can be described in very simple terms or even by one parameter – the Hurst exponent H.

In the case of computer systems the existence of long-term dependencies was discovered in 1992 when in Bellcore the simple experiment was done [19]. It turned out that computer networks traffic is governed by such dependencies. This discovery forces the technological changes in computer networks. However the problem of influence of long-range dependencies in the case of all aspects of computer systems is not widely discussed in literature. This is a very surprising fact because the implementations of Turing machines are physical systems and probably are quite often governed by such dependencies. As the example one can read [16] where it was shown that long-range dependencies in input data influence the number of dominant operations necessary to sort the input set by insertion sort algorithm. This problem is developed in Sect. 4.

3.2 Long-Range Dependencies – Spatial Domain

The another reason that *moves* computer systems towards complex ones is the problem of their structure in a spatial domain. Such a structure can be analyzed at least by two approaches. The first one is based on graphs: each graph node can represent one element of the system and the existing connections (dependencies) between nodes are represented by edges. In such a case the most important problem is the analysis of graph properties. It can quickly turn-out that such a graph is not random one but represents a structure that is rather a complex network (a "small world") [20]. Despite that the graph consists of huge number of nodes, the average distance l is small and the clustering coefficient C is greater than in a random graph. Moreover in this case one can use the different levels of particularity and see that such graphs can exist when the analysis of analogue and digital circuits is done [21], connections between computers in network are considered [22] or between links on www pages are analyzed [23].

[3] It means that in task there are not any known dependencies – the task represents the "white noise", i.e., a structure without long-range dependencies in time and space domain.

The second approach can be more *spectacular* – one can see for example the graphical visualization of the whole structure of software (for example the operating systems) with as many details as it is possible (functions, procedures, subprograms, etc.) [24]. Such an approach can show the software as a complex structure similar to fractal.

The statement that computer systems are heterogeneous ones is obvious. The list of companies that produce different components of computer systems is very long. But each company has got its own secrets and often introduces the different technologies. Each connection between these different technologies turned out to be a *botleneck* because the cooperation between them is possible when *the special translations* are done. For example let us consider the case of 4G technologies. The comparison of different 4G communication technologies (see Table 1) shows that the throughput can be from 9.7 Kb/s to 70 Mb/s (the dispersion about four orders of magnitude) while the range from 10 m to 35 km (the dispersion about three orders of magnitude) [25].

Table 1. 4G Telecommunications Technologies

Network	Area	Throughput	Cost
B-GAN	World	max 144 Kb/s	High
GSM/GPRS	< 35 km	9.6 Kb/s – 144 Kb/s	High
IEEE 802.16a	< 30 km	max 70 Mb/s	Medium
IEEE 802.20	< 20 km	1 − 9 Mb/s	High
UMTS	20 km	< 2 Mb/s	High
IEEE 802.11 g	100 − 300 m	54 Mb/s	Low
HIPERLAN 2	70 − 300 m	25 Mb/s	Low
IEEE 802.11a	50 − 300 m	54 Mb/s	Low
IEEE 802.11b	50 − 300 m	11 Mb/s	Low
Bluetooth	10 m	max 700 Kb/s	Low

It is well known (for example in MOS transistors) that the dispersion of time [26] and spatial constants [27] leads to the existence of excess $1/f$ noise, which in frequency domain is the manifestation of time long-range correlations. Such a dispersion is the main cause for the existence of statistical self-similarity; behavior of queues no longer can be characterized by a simple Kendal's notation, but becomes very dynamic and complex influencing service point performance [28]. The connections between each subsystem (especially in 4G technologies given in Table 1) require the existence of such queues with limited length of buffers in connecting nodes. If the transmission between two subsystems exceeds the node throughput some packets will be dropped and retransmission appears – the system becomes unstable and the high order phenomena govern its dynamic. This problem was discussed for example in the case of TCP protocol where in [29] it was shown that the dynamic behavior of this protocol can be represented by a bifurcation diagram.

3.3 Long-Range Dependencies – Stored Data

The so far presented examples consider the problem of spatial software and hardware structure and the problem of time long-range dependencies in processing. But one can also ask: what about the input data that are processed and stored in computer systems. It is well-known that texts in many languages follow the Zipf's law [30]. Because in computer systems one can store also in files the different texts it should be expected that such dependencies should exist in files [31]. It should be also noticed that input data can also consist of numbers and in such a case one might be aware of the existence Benford's law that describes the probability of occurrence the first significant digits in numbers [32]. In the case of computer systems such numbers are also the file sizes and they also follow this law – one can easily check this analyzing hard drives on own computers[4] (Fig. 1). This problem can be connected with data mining and used for searching of new patterns in sets of data especially in data warehouses. For example in USA Benford's law is used for fraud detection [33].

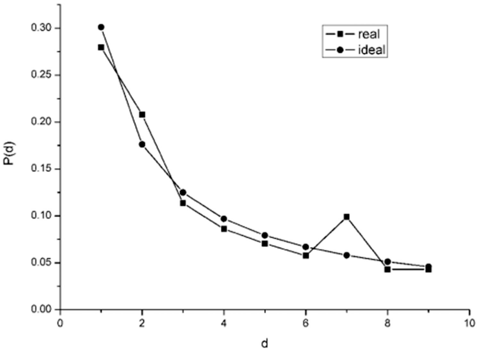


Fig. 1. Benford's law in author's personal computer file system. Line with dots – the ideal characteristics, line with squares – the real data obtained for author's desktop computers. There were about 250000 analyzed files. *d* – digit, *P*(*d*) – probability of occurrence digit *d* as a first significant digit

4 Physical Approach

The so far presented examples show that Turing idea given by its machine and the so far existed mathematical divagations about computer science can be quite far from real computer engineering. This problem comes from simple observation: computer science is a science that comes (directly) from math, but computer engineering is a part of our knowledge that rather comes from electronics, which is a physical science. Peter Wegner even said that [12]: "*computer science is a fundamentally non-mathematical discipline*". Nowadays computer science should be more closely connected with physics. The examples given above tell that properties of complex systems need

[4] This phenomenon has not been so far taken into account even in the case of SPEC Benchmarks.

physical approach for their explanation. But such an approach needs fundamental basis. For physics such a basis is a thermodynamics – the science that explains (almost) all problems with the energy flow and the efficiency especially when the dissipation occurs. Turing machine is a system that works as a mathematical model that does not need any energy, meanwhile, its each real implementation is a system that is made by elements working only when some amount of energy is given. Whenever exist the energy flow, the entropy problem also appears. In equilibrium thermodynamics the system stays in state where entropy is not produced any longer; quasi-equilibrium denotes the state when the entropy production is on the lowest possible level. In non-equilibrium states no one can tell how the entropy is produced [34]. So far the problem of entropy was described by Boltzmann-Gibbs definition of entropy given by famous equation $S = k \ln W$ (W denotes the number of equiprobable microstates), but this definition can be used only when the (quasi)equilibrium exists. However, in many situations there is no equilibrium and the physical basis of analyzed processes can be given by Tsallis definition of entropy [35]. Having such a definition of entropy one is able to give a better description of behavior of complex computer systems. The reason is very simple – this definition of entropy assumes that equilibrium state is one of the possible states (when entropic parameter $q = 1$). The whole problem is more complicated because the real computer systems are the open systems and they constantly exchange the entropy/energy with the environment. The simple example of above presented deliberations can be the thermodynamic analysis of insertion sort process. It is one of the simplest sorting procedures that works similarly to the behavior of bridge player who sorts his cards before the game [36]. In this algorithm exist two loops: the external one guarantees that all elements in the input set will be sorted and the internal one that is responsible for finding a right place for each currently sorted element (key).

The number of executions of this loop can be used to calculate the computational complexity, which in the worst case is $O(n^2)$, but in the best case is $\Omega(n)$ [36]. Only in these cases it is possible to say exactly how many dominant operations will be needed for each element, but among them there are also $n! - 2$ other cases for each instance that has got size n. Because the classical computation complexity describes only the total time needed for computation and does not say anything about how this time will behave for different instances (with the size n) from dynamical point of view it seems that it is not interesting for us. The worst and the best cases for this algorithm are deterministic, i.e., the number of dominant operations can be given by mathematical equation [36]. In other cases one cannot write such an equation, but they can lead to the dynamical behavior of this algorithm. The problem comes from assumption that in the classical computational complexity the intrinsic features of the task (i.e., the input data structure) are not being taken into account. In many cases input data structure can have very interesting properties and only in extreme cases it is pure deterministic or pure random like in the case of graphs, when the increasing degree of probability of edge existence between two nodes in random graphs can lead to the existence of *small worlds* or *scale free* property [20, 21]. So, the simple description of algorithm behavior by giving the computational complexity expressed by the big O notation seems to be insufficient especially if one realizes that the computation in reality is not made by Turing machines that are only mathematical models, but by nowadays computer systems that work simultaneously in the algorithmic and interactive mode [12].

Sorting is a procedure that introduces order into processed set – as a result of this kind of processing in output set we have less disorder, i.e., existing entropy is taken outside processed set. When the entropy production in the insertion sort algorithm is on the lowest possible level? Let's see the following way of thinking: n denotes a size of sorted set, n_i denotes the number of successively sorted elements ($n_i = 0 \ldots n$), M denotes a total number of executions of internal and external loops needed for sorting successive elements from the input set, M_1 denotes a number of executions of external loop – in insertion sort algorithm for each key it is equal 1; M_2 is the number of the internal loop executions and it can change from 0 (i.e., sorted key so far has got the maximum value) to $n_i - 1$ (i.e., sorted key so far has got the minimum value), M_3 stands for the number of such a possible internal loop executions that can appear but do not appear due to some properties of the input set. It is clear that $M = M_1 + M_2 + M_3$. To consider the thermodynamic properties of the sorting process for each sorting set of length n one needs to define a number W of possible configurations of loops executions that can appear during sorting the best, the worst and the other cases. This number is given as [37]:

$$W = \frac{M!}{M_1!M_2!M_3!} \tag{1}$$

For optimistic case it is $W_O = \frac{n_i!}{1!0!(n_i-1)}$, for pessimistic it is $W_P = \frac{n_i!}{1!(n_i-1)0!}$, whereas for the case where we need at least one excess dominant operation it is $W_D = \frac{n_i!}{1!1!(n_i-2)} = n_i(n_i - 1)$ and this number is greater than W_O and W_P showing that each case different than optimistic or pessimistic leads to entropy production bigger than in equilibrium state. Some details about these levels were considered in [37] where a direct connection to Tsallis entropy was proposed (for example the estimated value of Tsallis q parameter was: $q \in (1.15, 1.32)$), whereas in [38] it was presented how the total amount of entropy production during insertion-sort can be calculated. Obtained results show that this sorting is done in states far from equilibrium.

5 Conclusions

In paper it has been shown that there is a real need to do the paradigm change in computer science. The so far existing approach based on simple systems is not enough. The dependence of algorithm behavior on the data structure, the existence of long-range dependencies in time and spatial domain in computer systems, their physical limitations due to a need of energy dissipation and entropy production and new mode of working – interactive computing – are only a few important reasons that cause the different view on computer engineering. Presented examples and literature review can be a basis of new investigations and challenges in analysis, description, design and development of different computer systems, systems with computers, algorithms. They can be also used in computer science and engineering education especially if we realize that computers are omnipresent in our life and cause different revolutions, for example the idea of Industry 4.0. Modelling of such systems is possible when the concepts of complex scale-free networks, non-extensive statistics, power-laws, statistical self-similarity, long-range dependencies, fractals, high order phenomena, fractional calculus are taken into account.

References

1. von Bertalanffy, L.: General System theory: Foundations, Development, Applications. George Braziller, New York (1976)
2. Kuhn, T.S.: The Structure of Scientific Revolutions. University of Chicago Press (1962)
3. Amaral, L.A.N., Ottino, J.M.: Complex systems and networks: challenges and opportunities for chemical and biological engineers. Chem. Eng. Sci. **59**(8–9), 1653–1666 (2004)
4. Deutsh, K.: Mechanism, organism, and society. Phil. of Sci. **18**, 230–252 (1951)
5. Horakowa, J., Kelemen, J., Capek, J.: Turing, von Neumann and the 20th century evolution of the concept of machine. In: International Conference in Memoriam John von Neumann, pp. 121–135, John von Neumann Computer Society, Budapest (2003)
6. Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proc. London Math. Soc. Ser. **2**(42), 230 – 265 (1936). Errata appeared in Series **2**(43), 544–546 (1937)
7. Penrose, R.: The Emperor's New Mind: Concerning Computers, Minds, and The Laws of Physics. Oxford University Press (1989)
8. Gödel, K.: Uber formal unentscheidbare Satze der Principia Mathematica und verwandter Systeme, Monatshefte fur Mathematik und Physik, vol. 38, pp. 173–198 (2000). On formally undecidable propositions of Principia Mathematica and related systems I. (Translated by Martin Hirzel)
9. Eberbach, E., Goldin, D., Wegner, P.: Turing's ideas and models of computation. In: Teuscher, Ch. (ed.) Alan Turing: Life and Legacy of a Great Thinker, Springer, Heidelberg (2005)
10. Church, A.: An unsolvable problem of elementary number theory. Am. J. Math. **58**, 345–363 (1936)
11. Wegner, P.: Research paradigms in computer science. In: Proceedings of the 2nd International Conference on Software Engineering, San Francisco, California, pp. 322–330 (1976)
12. Wegner, P., Goldin, D.: Computation beyond turing machines. Comm. ACM **46**(4), 100–102 (2003)
13. Knuth, D.: The Art of Computer Programming, vol. 1–3. Addison-Wesley Innovations (1968, 1973, 1997)
14. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley (1994)
15. Mertens, S.: Computational Complexity for Physicists. Comp. Sci. Eng. **4**(3), 31–47 (2002)
16. Grabowski, F., Strzałka, D.: Dynamic behavior of simple insertion sort algorithm. Fund. Inf. **72**(1–3), 155–165 (2006)
17. Hurst, H.E.: Long-term Storage: An Experimental Study. Constable, London (1965)
18. Mandelbrot, B.B., van Ness, J.W.: Fractional brownian motions. Fractional Noises Appl. SIAM Rev. **10**, 422 (1968)
19. Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V.: On the self-similar nature of Ethernet traffic (extended version). IEEE/ACM Trans. Netw. **2**(1), 1–15 (1994)
20. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small- world' networks. Nature **393**, 440–442 (1998)

21. Ferrer i Cancho, R., Janssen, C., Sole, R.V.: Topology of technology graphs: small world patterns in electronic circuits. Phys. Rev. E **64**, 046119 (2001)
22. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the Internet topology. In: SIGCOMM 1999: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 251–262 (1999)
23. Adamic, L.A.: The small world web. In: Proceedings of the 3rd European Conference Research and Advanced Technology for Digital Libraries, ECDL (1999)
24. http://fcgp.sourceforge.net/lgp/index.html (2007)
25. Vidales, P., Baliosian, J., Serrat, J., Mapp, G., Stajano, F., Hopper, A.: Autonomic system for mobility support in 4G networks. IEEE J. Sel. A. Comm. **23**(12), 2288–2304 (2005)
26. Grabowski, F.: Influence of dynamical interactions between density and mobility of carriers in the channel on 1/*f* noise of MOS transistors below saturation I. Mechanisms. Solid-State Electr. **32**(10), 909–913 (1989)
27. Grabowski, F.: Difference between the 1/*f* noise spectral density before and after stress as a measure of the submicron MOS transistors degradation. Mic. Rel. **35**(3), 511–528 (1995)
28. Strzałka, D.: Influence of long-term dependences on hard drives performance during human computer interaction. Acta Phys. Pol. A **129**(5), 1064–1070 (2016)
29. Li, W., Zeng-zhi, L., Yan-ping, C.: A control theoretic analysis of mixed TCP and UDP traffic under RED based on nonlinear dynamic model. In: Third International Conference on Information Technology and Applications, vol. 2, pp. 747–750 (2005)
30. Zipf, G.K.: Human Behaviour and the Principle of Least-Effort. Addison-Wesley (1949)
31. Zebende, G.F., de Oliveira, P.M.C., Penna, T.J.P.: Long- range correlations in computer diskettes. Phys. Rev. E **57**(3), 3311–3314 (1998)
32. Benford, F.: The law of anomalous numbers. In: Proc. of the Am. Phil. Soc. **78**(4), 551–572 (1938). Nigrini, M.: A taxpayer compliance application of Benford's law. J. Amer. Tax. Assoc. **18**, 72–91 (1996)
33. Prigogine, I., Stengers, I.: Order out of Chaos. Man's New Dialogue with Nature. Bantam Books, New York (1984)
34. Tsallis, C.: Possible generalization of Boltzmann-Gibbs statistics. J. Stat. Phys. **52**, 479 (1988)
35. Cormen, T.H., Leiserson, Ch.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press (1994)
36. Tsallis, C., Levy, S.V.F., Souza, A.M.C., Mayanard, R.: Statistical-mechanical foundation of the ubiquity of levy distributions in nature. Phys. Rev. Lett. **75**, 3589 (1995)
37. Strzałka, D., Grabowski, F.: Towards possible non-extensive thermodynamics of algorithmic processing - statistical mechanics of insertion sort algorithm. Int. J. Mod. Phys. C **19**(09), 1443–1458 (2008)
38. Strzałka, D.: Selected remarks about computer processing in terms of flow control and statistical mechanics. Entropy **18**(3), 93 (2016)
39. Waldrop, M.: Complexity: The Emerging Science at the Edge of Order and Chaos. Simon and Schuster (1993)

Improving FPGA Implementations of BLAKE and BLAKE2 Algorithms with Memory Resources

Jarosław Sugier[✉]

Faculty of Electronics, Wrocław University of Science and Technology,
Janiszewskiego Street 11/17, 50-372 Wrocław, Poland
jaroslaw.sugier@pwr.edu.pl

Abstract. BLAKE is a cryptographic hash function which was one of the 5 finalists in the SHA-3 competition and although it ultimately lost to Keccak the algorithm was very well received for its both cryptographic strength and performance. In this paper we propose particular modifications in hardware implementation of the cipher which employ built-in FPGA block RAM modules in order to eliminate involved distribution of message bits among cipher rounds. The idea is tested on 4 different architectures: the standard iterative one and three loop-unrolled organizations with 2, 4 and 5 rounds instantiated in hardware. Parameters of the architectures implemented with two popular FPGA families – Spartan-3 and Spartan-6 – indicate that substantial reductions in design size can be achieved also with some (albeit not so spectacular) improvements in speed.

Keywords: BLAKE hash algorithm · Implementation efficiency · Block RAM · Resource utilization

1 Introduction

Although BLAKE eventually lost to Keccak in the SHA-3 competition the cipher is still often selected as a hash function of choice in contemporary data processing systems due to its excellent cryptographic strength and high efficiency in software. Potential of its hardware implementations, like of any other SHA-3 candidate, was extensively studied e.g. in [5–7]. In this paper we focus on one particular aspect of BLAKE hardware realizations, dealing with challenges caused by its specific peculiarity: the need of involved distribution of message bits among cipher rounds. Implementation of this distribution is much more cumbersome in hardware than in software and its elimination can significantly reduce FPGA utilization and improve overall performance. The proposed idea consists in replacing the distribution with repetitive storage of the message in block RAM modules located within every round instance. In the paper we test this solution in 4 different architectures of the cipher: the standard iterative one and three loop-unrolled organizations with 2, 4 and 5 rounds instantiated in hardware, for both BLAKE and BLAKE2 variants. The results found after their implementation in popular Spartan-3 and Spartan-6 devices from Xilinx are

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_38

compared to parameters of analogous architectures implemented without memory so that the savings in array utilization can be measured against supplementary cost of occupied block RAM modules.

The first results investigating potential of the proposed modification were presented in [12]. In this work we extend them with another FPGA platform and add the BLAKE2 variant of the algorithm. Contents of the paper is organized as follows. After briefly outlining the BLAKE hash functions in the next section, in Sect. 3 we present standard iterative and loop unrolled architectures viable for its implementation and introduce the proposed method of block RAM application. Finally, in Sect. 4 we discuss the results obtained after implementation of the modified architectures and evaluate them against known parameters of the analogous organizations without the modification – a total of 32 design cases is included in the comparison.

2 The Family of BLAKE Hash Algorithms

2.1 BLAKE

In this study we will consider BLAKE-256 and BLAKE2 s size variants of the ciphers i.e. those which produce 256b hash value, internally handling 32b words and 512b state.

In BLAKE-256 [1] the plaintext message m is padded so that its total bit length is a multiple of 512 and then it is split into 512b blocks m^0, …, m^{N-1}. The hash value h (m) is computed iteratively according to the HAIFA iteration scheme [4] with cumulative updates of the hash *chain value* h^i by *a compression function*:

$$h^{i+1} := compress\,(h^i, s, t^i, m^i), i = \ 0\ldots N-1 \qquad (1)$$

The other input parameters are s (*a salt*) – an optional 128b string provided for randomized hashing required e.g. in digital signature schemes, and t^i – number of message bits hashed so far. The first chain value h^0 starts with a constant pattern.

The compression function is the essence of the whole processing and is organized around *a state* - a 4 × 4 matrix of words $v_0 \ldots v_{15}$. The state is initially filled with the chain hash value h^i, the salt and the counter (but not with the m^i bits!) and then it goes through n_r = 14 rounds. In every round it is modified by 8 instances of so called *G function*: $G_i(a, b, c, d)$, i = 0…7, transforms a set of 4 state words (selection of the words depends on i and the round number) and it is defined as a sequence of bitwise xor operations, 32b additions and rotations, using as a side input the message words mixed (xor'ed) with *constants c*.

2.2 Modifications in the BLAKE2 Algorithm

The SHA-3 evaluation proved a very large security margin of the BLAKE algorithm and in 2013 the authors proposed its improved version – called BLAKE2 – with modifications aimed mainly towards its simplification and optimization [2]. In a brief summary the following changes were introduced (the 256-bit version of the algorithm is considered):

- number of rounds was reduced from 14 to 10;
- message padding was simplified and its functionality was partially replaced with *finalization flags* f_0 and f_1;
- initialization of the h^0 chain value was extended with *parameter block* (which includes, among others parameters, the salt and the finalization flags);
- the salt was removed from argument list of the compression function and was kept exclusively for the h^0 initialization;
- the constants c were dropped and the message words enter the G function without mixing.

For efficiency of hardware implementations investigated in this paper most of the above changes are of little significance: resources needed for both the h^0 and state initializations are negligible when compared to the hardware representing the actual compression in the eight instances of the G functions. Also the reduction in the number of rounds, while obviously cutting the number of clock cycles needed for completion of the computation, can be done with trivial adjustments in the control unit. Nevertheless, removing the c constants indeed simplifies realization of the G function in hardware: keeping a total of 16 words, each 32b wide, in multiple ROM modules and multiplexing them on the two inputs of each G instance is the main difference between the hardware of BLAKE and BLAKE2 realizations – although it is a relatively minor one looking on complexity of the remaining part of the function.

3 Implementing BLAKE in Hardware

3.1 Organizations of Data Processing

Processing scheme of the BLAKE algorithms is typical to any round-based cipher and it can be efficiently implemented in software in a CPU-based system in an iterative manner: operations of a single round are expressed in the code once and then applied to the state variables v_i repeatedly in a loop n_r times. When transferring the algorithm to hardware (either ASIC or FPGA) the designer is facing a larger diversity of feasible implementation options. In general, there are two opposite extreme approaches: the iterative loop of the cipher can be completely unrolled with all the rounds replicated in hardware as a cascade of n_r modules, or the loop is not unrolled at all with just one round module implemented in hardware and its operation on state signals is repeated n_r times (that is, in n_r clock cycles) in a manner resembling software iterations. Furthermore, as a mid-range solution the loop can be unrolled in part: one fourth, for example, of the rounds can be reproduced in hardware and the state signals are passed through them four times. In this paper, after universal taxonomy presented e.g. by Gaj et al. [6], an architecture with k unrolled rounds will be denoted as xk while the basic iterative one – as x1.

In this study we focus on high speed organizations and the following 4 organizations were selected for the test suite:

- x1: the basic iterative architecture with one round implemented in hardware and the state being passed though it repeatedly in 14 (BLAKE) or 10 (BLAKE2) clock cycles (i.e. each complete round is computed in one clock tick);
- x2: modification of the above with a combinational cascade of two rounds implemented in hardware with total computation done in 7 or 5 clock cycles (with each clock tick the state is propagated through two rounds);
- x4: the cascade is built from 4 rounds and 4 or 3 clock cycles are required for complete computation (in the last cycle the result is taken from the second round in the cascade in order to get $n_r = 14 = 3 \times 4 + 2$ or $n_r = 10 = 2 \times 4 + 2$);
- x5: as in the previous case but with 5 rounds unrolled in hardware; in BLAKE 3 clock cycles are needed for complete computation (the final result is taken from the fourth round in the cascade) while in BLAKE2–2.

3.2 Challenges of BLAKE Message Distribution

Although BLAKE follows the rules of in-round processing of ChaCha cipher [3], it introduces significantly different distribution of the message bits among the rounds. In ChaCha and in majority of other hash functions (including SHA-3 winner KECCAK) the message bits which enter the compression are routed only to the input to the first cipher round in parallel with other data like salt, counter or nonce, forming the initial value of the state. That is, the message bits enter only beginning of the round cascade and are not propagated to each round separately: after creating the initial state the message bits are not utilized afterwards. BLAKE, although its authors utilized ChaCha core transformations, uses a different approach: instead of being loaded at the input of the round cascade, the message words are sent to the *G* functions (two words per function) in each round. The data flow inside hardware implementation of the BLAKE's *G* function is presented in the left part of Fig. 1, where the two side entries for the message words illustrate the problem. For BLAKE2 the only difference would be in removing the $c_{[]}$ constants which are xor'ed with message words $m_{[]}$.

The authors even do not mention in the specification this change in utilization of the message data treating it as a relatively minor extension of the ChaCha processing scheme. Indeed, it may be so in software implementation but in hardware this means that the message bits must be routed separately to each *G* instance. This leads to creation of a completely new, 512b wide data path which has not been needed neither in ChaCha, Salsa20 nor in Keccak as we have analysed in our previous works [8, 11]. In effect this doubles the total width of the data path running along the round cascade from 512b (the state) to 1024b (the state plus the message bits). This configuration is illustrated in the right part of Fig. 1 taking the x4 case as an example.

Handling the message words is additionally complicated by word permutations $\sigma_0 \ldots \sigma_9$ used by the algorithm [1]. In each round a different permutation of m_i is used so switching between them requires supplementary multiplexers controlled by the round counter. In the x1 architecture every *G* instance reads 10 different pairs of m_i words thus in total 16 multiplexers 10:1 needed to be implemented but with loop unrolling the number of multiplexers increases proportionally with the xk factor while

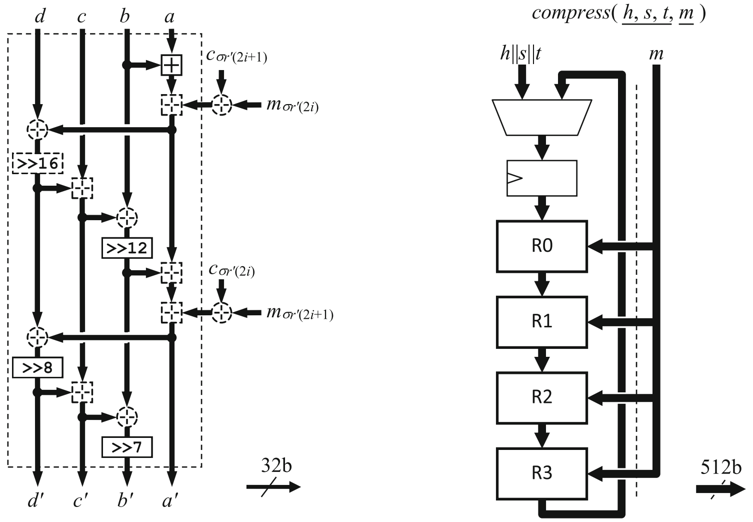


Fig. 1. Data propagation inside the BLAKE *G* function (left) and specific distribution of message bits in the x4 architecture (right).

their width is reduced: for example in the x4 organization the first two round instances apply four, and the two last ones – three σ permutations, and such will be the numbers of inputs in the multiplexers. More detailed discussion about permutations assigned to the round instances and the resulting distribution of the m_i words among the *G* functions in all the unrolled architectures is included in [10]. The multiplexers operating on 32b words do occupy substantial extra resources in the FPGA fabric.

3.3 Replacing Data Distribution with Memory Storage

In order to address the problems of message distribution we propose taking an advantage of block RAM modules available in the FPGA chips which constitute the implementation platform. The main idea [12] is to assign one such module for each instance of the *G* function implemented in hardware so that the involved dissemination and multiplexing of the *m* bits is avoided. Figure 2 compares realization of the four *G* functions as they are realized in hardware (creating complete half of the round) in a standard way without RAM and with the proposed extension.

Xilinx Spartan FPGA devices which are used in this study offer as additional resources complementing the programmable array so called block RAM modules and these were used for the purpose of message storage. Each module can store 16 kb of data and can be configured with different depth vs. width ratios – in organizations from 16 k × 1b to 512 × 32b. For our application the last case – 512 × 32b – is a suitable one with each m_i word occupying exactly one memory location. Additionally, the modules offer full dual port functionality, i.e. their contents can be simultaneously accessible (both for reading as well as for writing) through two equivalent ports.

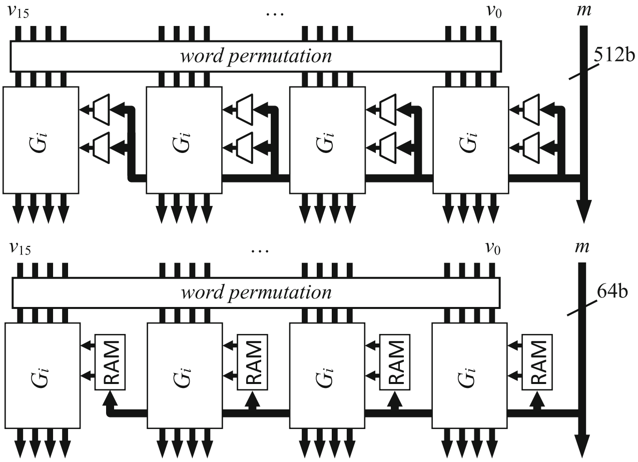


Fig. 2. Distributing and multiplexing the message bits to all G modules in the standard implementations (above) versus storing them in dual-port RAM units assigned to each module (below).

This feature is ideally suited to the needs of the proposed mechanism: one module can concurrently read two m_i words in one clock cycle as they are required for computation in one G function, and the total number of modules can be reduced by half compared to application of single-port memories. Still, the number of utilized modules is relatively high: every complete round in hardware needs 8 RAM units so their final number in the investigated architectures range from 8 (the case of the x1 organization) to 40 (x5). These figures should be compared to the total of 104 and 268 block RAM units offered by the particular Spartan-3 and Spartan-6 chips selected in this work for implementation tests. Also, utilization of RAM capacity is quite low: of the total 512 cells in each just 16 (1/32) are actually taken by the complete message.

All the RAM modules must duplicate the same message and they must be loaded with this data before the actual computation begins. The loading introduces obligatory initialization phase which adds extra delay and in some cases can remarkably slow down the total execution time. Nevertheless, thanks to the dual port interface two message words can be loaded in parallel so the loading operation needs 8 clock cycles and these cycles can be much shorter than the ones required during actual computation which starts afterwards.

Further issues arise regarding memory synchronization. The block RAM is a fully synchronous module also in read operation, i.e. when the read address is established the read data appears on the outputs only after the clock edge. This means that without appropriate compensation in clock cycles when computing some particular round number i the RAM outputs would present message words for the previous round $i - 1$ – in order to solve this problem one void clock cycle is needed to “precharge” RAM outputs. In BLAKE, where m words are mixed with c constants, the counters used for reading σ permutations of m words need to be one cycle ahead of those used for addressing the c constants so the two sets of counters are needed. In BLAKE2, as there is no need to address the *c* constants, the counters do not need to be doubled. All in all, together with the 8 clock cycles needed to load message data to RAM modules the preliminary phase adds in total 9 clock cycles before the actual computation of the compression can start.

4 Results

4.1 Parameters of the Implementations

Both versions of the cipher were implemented in all four architectures in configurations where the main hardware module computing the compression function was equipped with basic input/output registers providing means for iterative hashing of the message in 512b chunks. Then the eight designs were automatically synthesized and implemented by Xilinx ISE software with XST synthesis tool for two devices – Spartan-3 XC3S5000-5 [13] and Spartan-6 XC6SLX150-3 [14]. These chips were selected because they are sufficiently large to accommodate even the most sized x5 organizations. The same approach was applied in our previous works on BLAKE [8, 9] so an already existing test platform was uniformly extended to accommodate BLAKE2 version, keeping the ability to produce comparable results.

The results obtained after implementation of the two ciphers without and with RAM, in all 4 organizations, on both hardware platforms – a total of 32 test cases – are presented in Table 1. Speed aspect is represented in the first column of each group by the value of the minimum clock period as it was estimated after static timing analysis of

Table 1. Implementation results

		Min. T_{clk} [ns]	Levels of logic	Routing delay [%]	Size: LUTs	Size: Slices	Min. T_{clk} [ns]	Levels of logic	Routing delay [%]	Size: LUTs	Size: Slices
		BLAKE					BLAKE RAM				
Spartan-3	x1	45,7	66	50,6	9155	5415	40,3	62	44,4	4961	2860
	x2	88,9	118	51,2	16928	10039	83,6	100	57,1	8684	4894
	x4	189,7	203	58,7	32933	19000	157,7	180	53,0	16142	8638
	x5	244,1	258	61,7	41923	23232	197,3	229	53,3	20448	10913
Spartan-6	x1	28,7	35	64,3	5621	2460	26,4	43	56,5	3187	1024
	x2	67,7	70	72,3	10150	4409	51,9	80	61,7	5646	2501
	x4	108,3	150	65,6	18994	8396	106,8	119	68,4	10401	4790
	x5	185,1	131	78,1	24368	8833	116,3	179	60,1	12759	5184
		BLAKE2s					BLAKE2s RAM				
Spartan-3	x1	43,4	29	59,7	8370	4995	40,0	66	41,3	3935	2353
	x2	86,7	85	58,7	15058	8620	78,0	119	46,2	6863	3809
	x4	195,7	209	59,9	27906	16549	151,9	219	47,6	12396	6576
	x5	246,7	250	61,1	34492	20065	194,9	267	50,2	15426	8254
Spartan-6	x1	27,2	28	69,1	4856	1544	25,8	38	62,8	2762	797
	x2	62,4	74	68,9	8534	3080	55,7	70	67,0	4654	1529
	x4	150,6	134	76,0	16168	5124	100,5	145	63,5	8396	3698
	x5	171,4	159	74,8	19777	6479	146,9	169	69,7	10267	3944

the final, fully routed design. The next two columns provide parameters which illustrate effectiveness (or difficulties) of the implementation process, i.e. how the complex logical transformations of the algorithms were realized with programmable resources of the array: for the longest combinational path in the design the second column gives number of logic elements it contains and the fourth – percentage of the propagation delay incurred by the routing resources (and not logic elements). Any significant rise in the latter parameter above 50–70% indicates problems with routing of connections between logic elements of the array: if the connections are too dense vs. distribution of logic elements, routing becomes congested and implementation in the array becomes problematic. Size characteristics are reported in the last two columns which give the total numbers of utilized LUT generators and slices.

4.2 Advantages of Applying RAM Modules

Based on the data from Table 1 we can evaluate size and speed of implementations modified in a way proposed in this paper and compare them against the results of the standard approach without RAM utilization. Parameters of those standard implementations for BLAKE algorithm are taken from previous studies in [8, 9] while the results for BLAKE2 are presented here for the first time. Such a comparison is the purpose of Fig. 3: the minimum clock period T_{clk} and the number of LUT generators for implementations with RAM are expressed as percentages of corresponding values of standard (no RAM) realizations – for all four organizations, for both versions of the hash, and for the two FPGA platforms (hence such a number of data on display). As one can see, in all cases the percentages are below the 100% level, i.e. the RAM implementations were faster (shorter T_{clk}) and smaller than their traditional counterparts.

While it was obviously expected that moving part of the logic from the FPGA array to the block RAM should reduce LUT utilization, the actual scale of this reduction is outstanding: on average LUT numbers were cut by half, in the BLAKE2 implementations on Spartan-3 even to 47–44%. Reductions on Spartan-6 platform are on average smaller by 5–10% than on Spartan-3 – albeit still to the level of 57–52% – also the original BLAKE hash is improved by somewhat smaller amount then the BLAKE2 variant. But the scale of the improvements indicates what burden was placed on the implementation tools when the 16 message words, 32b each, are to be delivered and selected in multiplexers twice in each *G* module: this task alone took approximately half of the designs with only rest of the resources busy with actual arithmetic of the hash (which – as in any other cryptographic algorithm – is a very complex job on its own).

The improvement, although not so stable across all configurations, is seen also in performance characteristic: while reading the message words from the block RAMs does introduce some delay, passing them through the distributed logic in the standard architectures turned out to be even slower so the overall clock period is reduced on average by 15–25%. But first of all, when looking at the T_{clk} parameter we should note worse stability across the unrolled architectures on the Spartan-6 platform. The newer family, although can offer significantly larger chip capacities and better performance, may suffer from routing congestion in case of more sized loop unrolled designs as it also indicated in previous analyses of other contemporary ciphers like Salsa20,

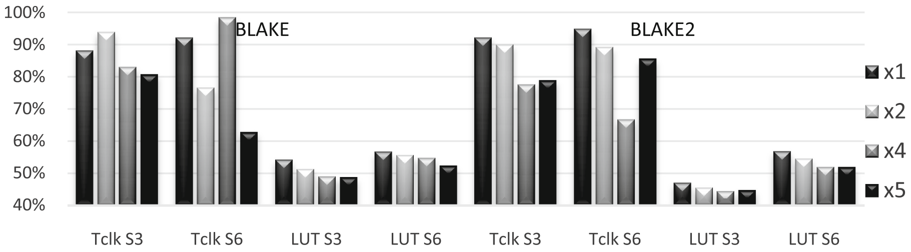


Fig. 3. The proposed RAM-based vs. standard approach: speed (minimum clock period) and size (number of function generators) of the implementations which use RAM as percentages of the values obtained previously, in Spartan-3 (S3) and Spartan-6 (S6) devices.

Keccak, AES or Serpent [8–11]. These problems can lead to less predictable results with specific implementations reaching better or worse results randomly, depending on particular effectiveness of routing optimization in the specific chip. Such instabilities were observed here for x4 and x5 implementations of BLAKE and for x4 case of BLAKE2 but they should not hinder the overall improvement of the investigated modification which was brought also on the Spartan-6 platform. Finally, one should note that routing glitches affect the propagation delays and can be seen in degraded T_{clk} values of the flawed designs but they have no influence on their mapping and placement, so the size characteristics – number of LUT elements – remain unaffected.

As other parameters from Table 1 confirm, with introduction of the RAM units the interconnection network gets simplified (fewer levels of logic) and is better routed (smaller percentage of longest path delay generated by interconnections). Specific individuality is observed in the case of the x2 design of BLAKE with the RAM extension on Spartan-3: in this particular configuration the optimization procedures especially efficiently reduced levels of logic but this was accompanied with a disproportional increase in the routing part of the longest path (the only case when an increase in any parameter is observed) so the reduction in the overall clock period is actually the smallest across all the architectures on the Spartan-3 platform.

To balance these positive effects on speed and size of the designs it should be reminded that they are accomplished at the cost of additional utilization of block RAM units. Strictly speaking, they require from 128 kb (the x1 architecture) to even 640 kb (the x5 case) of extra RAM in the hardware budget, even though only 1/32 of it is actually used. In case of ASIC implementation size of the RAM units can be tailored exactly to the needs of individual *G* functions. The detailed analysis (see [10]) indicates that the required net amount of memory ranges from 3.5 kb in the x1 case to 6.5 kb in the x4 organization.

4.3 BLAKE2 vs. BLAKE

The second evaluation will compare size and speed of BLAKE2 implementations versus the analogous BLAKE ones. Like in the previous comparison, to avoid analysis of raw numbers, Fig. 4 presents the ratios of respective parameters between two versions of the cipher, implemented with memory and in a standard way.

Once again an opening remark should be made about instabilities in speed (but not in size) of x4 and x5 cases on the Spartan-6 platform – filtering out these two T_{clk} S6 cases the analysis becomes more conclusive. Obviously, vast majority of the ratios is below 100% what means that BLAKE2 implementations were smaller and faster than the BLAKE ones but the scale of the improvement varies. The proposed RAM extension on the Spartan-3 platform offers average reduction in size to 78% which is the best outcome in this comparison; size optimizations for the standard (no RAM) implementations are on both platforms worse by about 5% than in RAM versions, but still down to 81–91%. Adding RAM helped in optimizations introduced in BLAKE2.

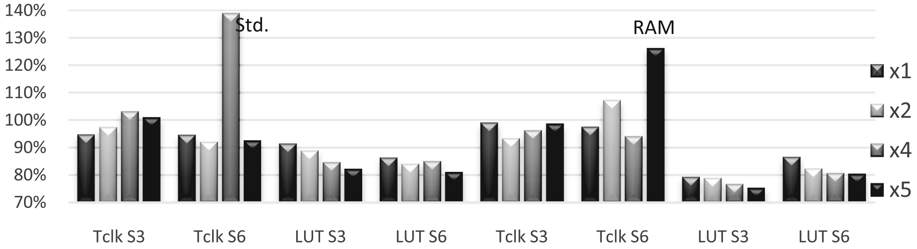


Fig. 4. Speed and size parameters of BLAKE2 implementations as percentages of equivalent BLAKE cases, in Spartan-3 (S3) and Spartan-6 (S6) devices.

Speed improvements are not as good and sometimes even questionable: although the standard implementations on Spartan-6 (with an exception of the aforementioned x4 case) reduce T_{clk} on average to 93%, the advantages in the RAM versions are very small at all. This shows that simplifications in *G* functions indeed helped in speed efficiency of the standard approach and introducing the RAM did not offer significant further improvements.

4.4 Scaling with the Loop Unrolling Factor

In order to consistently evaluate efficiency of the loop unrolling mechanism among diversity of ciphers variants and platforms, the analysis will be performed according to an approach similar to that proposed in [11].

In every cipher/organization/platform combination the x1 architecture will be taken as a point of reference and its parameters will be used for estimation of size and speed of the unrolled architectures in the following way. Size of each unrolled architecture should increase proportionally to the number of rounds implemented in hardware (the unrolling factor) and we will estimate

$$Size_{\mathrm{x}k} \approx Size_{\mathrm{x}1} \cdot \; k \tag{2}$$

Maximum frequency of operation – or the minimum clock period – depends on the other hand on the number of rounds the state must go through in one clock cycle, i.e.:

$$Tclk_{xk} \approx Tclk_{x1} \cdot \; k \tag{3}$$

The two above equations can be used in estimating T_{clk} and number of LUTs in the x2–x5 cases and Fig. 5 includes this kind of results displaying the ratios *actual_value/estimation* for both versions of the ciphers. The lower the ratio, the faster (shorter T_{clk}) or the smaller (lower number of LUT) was the actual design in comparison to what could be expected from its x1 case. The value of 100% is the threshold separating “better than” (<100%) from “worse than” (>100%) the expected.

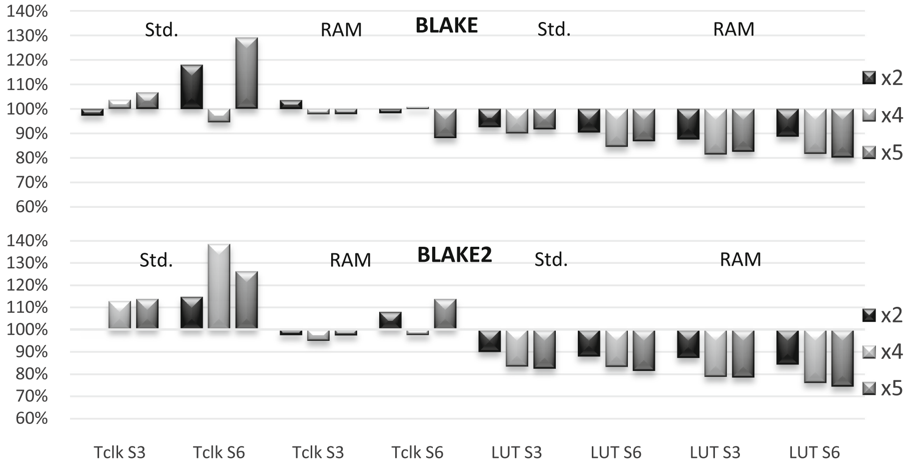


Fig. 5. Scaling efficiency of the unrolled architectures in BLAKE (above) and BLAKE2 (below): actual speed and size parameters are given as percentages of the values estimated from the x1 case.

Regarding the speed, it should be noted that only RAM implementations in the Spartan-3 devices of both cipher variants behave close to the expectations and the deviations from the estimated values are within ±5% margin. On this platform the standard (no RAM) implementations exhibit more problems in the BLAKE2 version where the x5 case is 14% slower than expected which shows that introducing RAM is more important for implementation efficiency than optimizations of BLAKE2. Again, speed parameters on Spartan-6 platform are less uniform but it can be seen that the standard (no RAM) implementations for both ciphers can be even 30–40% slower than the estimations while using RAM can reduce this amount by half.

Results of the size scaling are more consistent and the unrolled architectures are always smaller than estimations. The best reductions down to 75% are noted for RAM-based BLAKE2 cases in Spartan-6 chips and the smallest – for BLAKE implemented without RAM in Spartan-3 but the variety across all 8 cipher/variant/platform combinations is not as wide as it was in the speed parameter.

5 Conclusions

Although the core of BLAKE hash functions was developed as a modification of ChaCha algorithm, the additional data paths carrying the message bits to individual round instances (which are not required in ChaCha, Keccak and other contemporary ciphers) substantially complicate organization of the hardware implementation, increase size of the design and impair its performance. This paper discussed one solution of this problem: application of on-chip FPGA block memory which repetitively duplicate storage of the message words and provide them individually to every instance of the *G* function.

The idea was tested for both BLAKE and BLAKE2 ciphers on 4 different organizations: the standard iterative one and three high-speed loop-unrolled architectures with 2, 4 and 5 rounds instantiated in hardware. Results found after their implementation in popular Spartan-3 and Spartan-6 devices from Xilinx showed that the modification remarkably enhanced size of all the tested architectures: on average, occupation of the FPGA array was reduced by half and some improvements, albeit not so spectacular, were observed also in performance.

An open issue remains whether these improvements compensate the extra cost of memory blocks added to the design. Nevertheless, it is common in the FPGA practice that the whole design do not use all resources of the selected device and we have shown that in such situations, if there remain some free memory blocks in the chip, using such "leftovers" for improvements in BLAKE implementation is definitely an option worth consideration.

References

1. Aumasson, J.P., Henzen, L., Meier, W., Phan, R.C.W.: SHA-3 proposal BLAKE, version 1.3. https://www.131002.net/blake/blake.pdf. Accessed Mar 2017
2. Aumasson, J.P., Neves, S., Wilcox-O'Hearn, Z., Winnerlein, C.: BLAKE2: simpler, smaller, fast as MD5. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) Applied Cryptography and Network Security, ACNS 2013. LNCS, vol. 7954, pp. 119–135. Springer, Heidelberg (2013)
3. Bernstein, D.J.: ChaCha, a variant of Salsa20, 20 January 2008. http://cr.yp.to/chacha/chacha-20080128.pdf
4. Dunkelman, O., Biham, E.: A framework for iterative hash functions: Haifa. In: 2nd NIST Cryptographich Hash Workshop, vol. 22 (2006)
5. Gaj, K., Homsirikamol, E., Rogawski, M., Shahid, R., Sharif, M.U.: Comprehensive evaluation of high-speed and medium-speed implementations of five SHA-3 finalists using Xilinx and Altera FPGAs. In: The Third SHA-3 Candidate Conference, Available: IACR Cryptology ePrint Archive 2012, vol. 368 (2012)
6. Gaj, K., Kaps J.P., Amirineni, V., Rogawski, M., Homsirikamol, E., Brewster, B.Y.: ATHENa – automated tool for hardware evaluation: toward fair and comprehensive benchmarking of cryptographic hardware using FPGAs. In: 20th International Conference on Field Programmable Logic and Applications, Milano, Italy (2010)

7. Junkg, B., Apfelbeck, J.: Area-efficient FPGA implementations of the SHA-3 finalists. In: 2011 International Conference on Reconfigurable Computing and FPGAs (ReConFig), pp. 235–241. IEEE (2011)
8. Sugier, J.: Implementation efficiency of BLAKE and other contemporary hash algorithms in popular FPGA devices. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Proceedings of the 11th International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. AISC, vol. 470, pp. 457–467. Springer, Heidelberg (2016)
9. Sugier, J.: Implementing SHA-3 candidate BLAKE algorithm in field programmable gate arrays. J. Pol. Saf. Reliab. Assoc. **7**(1), 193–200 (2016)
10. Sugier, J.: Memory resources in hardware implementations of BLAKE and BLAKE2 hash algorithms. J. Pol. Saf. Reliab. Assoc. (submitted for publication)
11. Sugier, J.: Popular FPGA device families in implementation of cryptographic algorithms. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Theory and Engineering of Complex Systems and Dependability, Proceedings of the 11th International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. AISC, vol. 365, pp. 485–495. Springer, Heidelberg (2015)
12. Sugier, J.: Simplifying FPGA implementations of BLAKE hash algorithm with block memory resources. Procedia Eng. **178**, 33–41 (2017)
13. Xilinx, Inc.: Spartan-3 Family Data Sheet. DS099.PDF. www.xilinx.com. Accessed Mar 2017
14. Xilinx, Inc.: Spartan-6 Family Overview. DS160.PDF. www.xilinx.com. Accessed Mar 2017

Assurance Case Patterns On-line Catalogue

Monika Szczygielska[1] and Aleksander Jarzębowicz[2](✉)

[1] Ośrodek Badawczo-Rozwojowy Centrum Techniki Morskiej S.A.,
Dickmana 62, 81-109 Gdynia, Poland
monika.szczygielska@ctm.gdynia.pl

[2] Department of Software Engineering, Faculty of Electronics,
Telecommunications and Informatics, Gdańsk University of Technology,
Narutowicza 11/12, 80-233 Gdańsk, Poland
olek@eti.pg.gda.pl

Abstract. Assurance case is an evidence-based argument demonstrating that a given property of a system (e.g. safety, security) is assured. Assurance cases are developed for high integrity systems, as in many industry domains such argument is explicitly required by regulations. Despite the fact that each assurance case is unique, several reusable argument patterns have been identified and published. This paper reports work on development of an on-line assurance case patterns catalogue available in NOR-STA web-based software tool. This work included an extensive literature search, critical evaluation of available patterns and selection of most relevant ones, finally translation of selected patterns to their target representation. The paper also describes a validation case study in which an assurance case for medical devices was reviewed and restructured by introducing patterns. The resulting catalogue was published and its 45 patterns can be directly used in assurance cases built using NOR-STA tool.

Keywords: Assurance case · Safety case · Pattern · Catalogue

1 Introduction

Assurance case is "a structured set of arguments and a body of evidence showing that an information system satisfies specific claims with respect to a given quality attribute" [1]. Assurance cases demonstrating properties like safety or security are developed for high integrity systems and in many industry domains they are explicitly required by regulations [2–4]. Despite the fact that each assurance case is unique, several repeatable problems and situations can be identified. Such problems and generalized solutions for them can be described in the form of patterns. Patterns are used in other engineering domains (e.g. design patterns in software engineering [5]) as general reusable solutions to a commonly occurring problems within a given context.

Patterns have been adopted and successfully used for assurance cases [6]. Several reusable patterns have been published, but their descriptions are distributed among many sources, they are not uniform and their direct use in software supporting tools is usually not possible. The work reported in this paper was the implementation of the idea to create a unified pattern catalogue and to publish it in the Internet. The catalogue is supposed to include all relevant patterns, ready to use in a supporting tool.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_39

In Sect. 2 we outline the background of our research, which includes two main parts: in Sect. 2.1 we describe assurance cases, their usage and main notations; in Sect. 2.2 we provide brief outline of patterns concept and their application to assurance cases. Section 3 presents our contribution: the development and validation of pattern catalogue. In Sect. 4 we discuss conclusions and possible directions of future work. No separate "Related Work" section is defined, to avoid redundancy, as catalogue development (Sect. 3) included a literature review.

2 Background

2.1 Assurance Cases

Assurance cases are developed for high integrity systems to demonstrate that their particular quality attributes are achieved. The most common are safety cases [7] which provide arguments that a system is acceptably safe to operate i.e. will not result in harm to its environment. Other kinds of assurance cases include e.g. security cases, reliability cases or maintainability cases [8].

A growing demand for assurance cases can be observed in several industry domains. Regulations for automotive [2], railway [3], healthcare [4] explicitly require or strongly recommend issuing an assurance case for high integrity systems to be approved by a regulatory body. Also, assurance cases were recently addressed by recognized standards issued by ISO/IEC [9] (later adopted by IEEE) and by OMG [10].

Assurance case development begins with defining high-level claims about system's quality attribute(s). Then a supporting argument is provided. Such argument will include its own, more detailed, lower-level claims, which in turn need to be supported. When necessary, evidence is referenced in the argument.

As a simplified example, consider a top-claim stating that "*A system is safe to operate in its environment*". The supporting argument could include sub-claims that "*Hazard identification activity uncovered all potential hazards*" and that "*All hazards have been eliminated*". "*All hazards have been eliminated*" is further decomposed to sub-claims addressing particular hazards: "*Hazard A is eliminated*", "*Hazard B is eliminated*". Evidence referenced in such argument would include description of hazard identification process, resulting list of hazards, system design documentation etc.

This example is very simple, while the real high integrity systems are usually complex, include a number of components and integrate parts engineered using different technologies. As result, an assurance case for such system is also very complex and supported by a large number of evidence sources. Development and maintenance of real-life assurance cases require suitable ways of expressing argument structures and software tools providing adequate support. To address such needs, dedicated assurance case notations were designed, which allow to express the structure of assurance argument with all essential aspects. On a closer look, the above example lacks many important details e.g.: what is the context of a given claim (e.g. how a "hazard" is defined) or what argumentation strategy is used and what is the rationale behind it (e.g. why are the claims about identification of hazards and their elimination sufficient to argue that the system is safe). Assurance case notations capture such issues as its elements/building blocks.

The notations currently used include CAE (Claim-Argument-Evidence) [11], GSN (Goal Structuring Notation) [12] and NOR-STA [13].

Our work is part of the research done at Gdańsk University of Technology. This research on assurance cases (dated back to 2001 [14]) resulted in TRUST-IT methodology for assurance case development. NOR-STA notation depicted in Fig. 1 is the main component of TRUST-IT (arrows from A to B denote that element A can support element B). Another result is NOR-STA tool, an Internet based software which supports development and maintenance of assurance cases. We use both NOR-STA notation and tool in our work reported in the next sections.

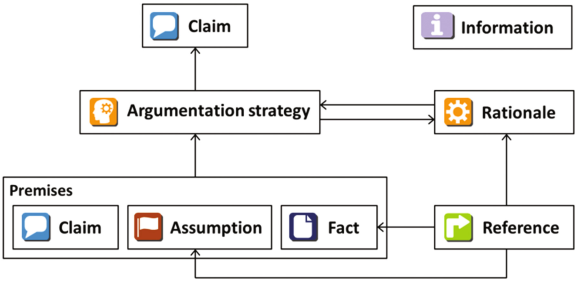


Fig. 1. NOR-STA notation metamodel

2.2 Patterns

Patterns are supposed to enable reuse of successful approaches by describing repeatable design problems and their generic (free of detail) solutions. Patterns were introduced to software engineering domain [5], where they became one of the most important ideas. Software design patterns capture repeatable problems related to object-oriented design and their optimal (with respect to flexibility, ease of maintenance etc.) solutions. According to [5], pattern‘s description should include the following elements:

Pattern Name, Intent, Also Known As, Motivation, Applicability, Structure, Participants, Collaborations, Consequences, Implementation, Example Applications, Known Uses, Related Patterns

Experiences from software design were adopted to assurance cases, based on the observation that repeatable problems and good practices can be identified for arguments as well. E.g. when demonstrating the safety of the system by using hazard analysis results (as in example from Sect. 2.1), an appropriate hazard decomposition pattern can be used. The first idea of assurance case patterns were described in [15] and the first catalogue of patterns was proposed in [6]. In the following years the concept and applications of patterns were further elaborated and more patterns (as well as catalogues grouping them) were published. Moreover, patterns were accepted by practitioners and currently are widely used in various industry domains [16–18].

3 Assurance Case Patterns Catalogue

3.1 Motivation

The main idea behind our activities was to develop and publish a catalogue summarizing the current state of research and practice on assurance case patterns. According to our prior knowledge from research on assurance cases, further confirmed by an initial literature review, the situation in assurance case patterns area was as follows:

- Several pattern catalogues existed (e.g. [6, 19]), but each of them included a set of distinct patterns (very few patterns shared), and some were dedicated to a more specific problems (e.g. systems built from existing software components [19]).
- Additional patterns, not included in any catalogue but described in separate papers or reports (e.g. [17, 20]) were designed. Such sources described a single pattern or a small number of interrelated patterns.
- Different notations were used for published patterns. The dominating notation used was GSN, but there were exceptions. Furthermore, pattern descriptions used different structures and some provided significantly more details than others.
- The published patterns were usually available only as the contents of a paper, thesis or report. Even if the author designed a pattern using a supporting software tool, such electronic pattern representation was not available to the general research community. Someone who intended to use a given pattern in his/her case, would have to manually enter it, element after element, into a software tool.
- Finally, as far as we know, in recent years no extensive search for existing patterns was performed and no summary about the current state published, except [21] where several sources and patterns known to authors are listed.

These observations motivated us to start the work aimed at development of a single pattern catalogue, summarizing the current state of this research area. We intended to represent those patterns in NOR-STA tool, so they could be published in the Internet and ready to be applied to assurance cases created in NOR-STA. In the next sections we describe the process of catalogue development, the resulting catalogue and the validation case study conducted to assess its applicability to support the work on assurance cases.

3.2 Catalogue Development Process

The first step to develop a pattern catalogue was to conduct a literature search and to identify sources where assurance case patterns are described. We cannot claim it was a Systematic Literature Review, as no review protocol was prepared and no meta-analyses conducted, but we made an effort to make the search as extensive as reasonably possible and to include all possible candidate sources. The main tool used for this purpose was Google Scholar (GS). Our experiences from past literature searches have shown it to be sufficient, as GS indexes other publication databases. This was also confirmed during this search – we reached, through GS, publications stored in other databases e.g. IEEE, Springer or Elsevier. We conducted the search by:

- Using appropriate keywords in GS search engine;
- Using citation maps for sources already found or known beforehand – for each source we checked its References sections and we used GS to identify publications which in turn cite this source;
- Using names of authors of previously identified publications in GS and in a generic web search engine.

The candidate sources found were analyzed by reviewing their titles, keywords and abstracts, also by quickly scanning the contents - pattern definitions are usually represented as figures. After rejecting the sources which clearly included no patterns, we still had 31 sources for a more thorough analysis. Due to space limitations we cannot list all of them in this paper, however such list of references is available in our on-line catalogue. The analysis of sources' contents resulted in several observations:

- Some of the sources, despite using word "pattern", just reported arguments used in development of a particular assurance case, not patterns understood as more generic, well-designed solutions, applicable to many assurance arguments.
- Other sources proposed patterns dedicated to a single domain (e.g. automotive [17] or nuclear [18]) or at least it was not clear whether they could be adopted to another domain (and if so, how should such pattern be modified/generalized).
- In a very few cases, a pattern appeared in more than one catalogue (e.g. ALARP Pattern in [6, 22]).
- When comparing patterns, some of them could be treated as more generalized versions of others (e.g. Architectural Decomposition pattern from [23] is more general than the corresponding patterns from [17]).

We made a selection among "candidate patterns" by applying the following actions:

- Reject the particular arguments not generalized into patterns.
- Reject patterns described as ideas only, without explicit argument structure.
- If a given pattern is included in multiple catalogues (or other sources) – select the most recent source.
- If similar argumentation structures are described as patterns – select the more general one.
- Reject domain-specific patterns, for which no indication is provided how to apply them to other domains.

As result, we selected 45 patterns. The most difficult decision concerned COTS Safety Patterns [19], we finally decided to include two core patterns and leave out the remaining ones.

The next steps were to translate these patterns into NOR-STA notation, to provide a uniform description structure for them and to represent them in NOR-STA tool. As all the selected patterns were defined in GSN notation, we had to define translation rules between GSN and NOR-STA notations. The translation encountered no fundamental problems, however some difficulties stemming from notational differences were uncovered e.g.:

- In GSN a Goal (an equivalent of a Claim) can be directly supported by a sub-goal. In NOR-STA an Argumentation strategy is mandatory between Claims to explain the argument. In such cases we had to add Argumentation strategies.
- In GSN, a Justification is an optional element. In NOR-STA, every Argumentation strategy is expected to have a Rationale, which had to be added.
- No explicit Context element is defined in NOR-STA, instead Information elements are used to express all contextual or explanatory information included in assurance arguments.
- GSN defines additional elements, which are useful for pattern instantiation e.g. marking some parts of an argument as optional or alternative. With no direct equivalent in NOR-STA, we had to use Information elements for this purpose.

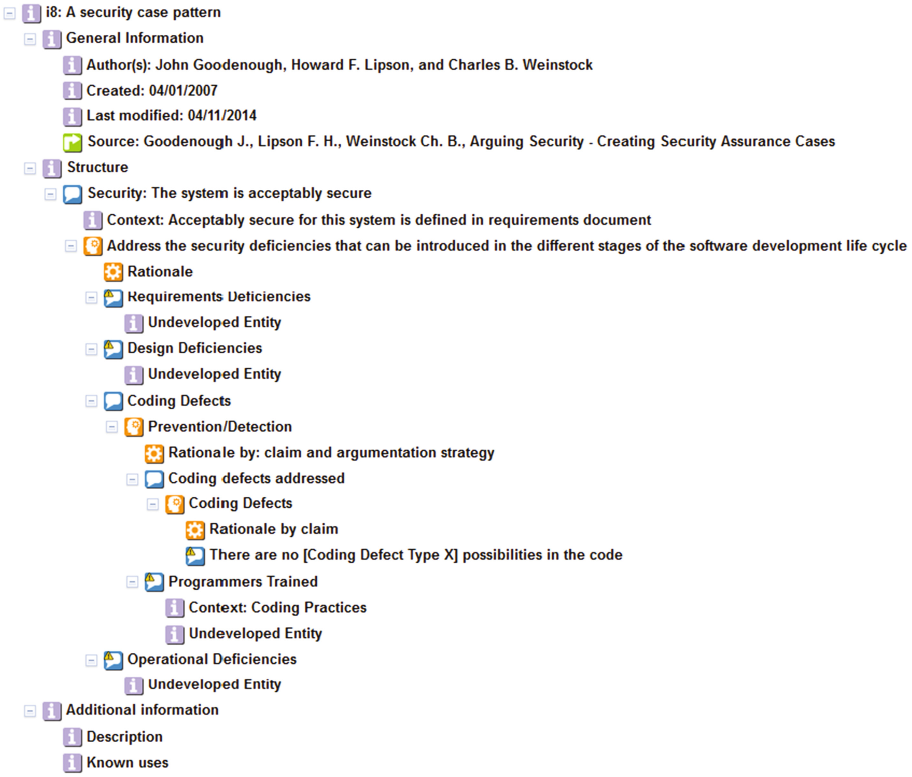
As for description structures, there were significant differences between the sources of selected patterns. Sources [6, 19, 24] used a full description structure adopted from software design patterns (listed in Sect. 2.2), [25] covered about a half of description elements, [22, 23] provided only a structure of the pattern, while [20, 26] a structure plus examples of use. All selected patterns were (manually) entered into NOR-STA tool, using translation rules mentioned above. All description elements defined for a given pattern were preserved. Moreover, for each pattern we provided a link to the source of its original description.

Figure 2 is a screenshot from NOR-STA tool depicting an example pattern (Security Case Pattern) as represented in our catalogue. Table 1 gives the summary of all patterns included in our catalogue and their sources. The catalogue is available on-line at [27] (please note that read-only access is enabled for unauthorized users). The total number of basic elements the included patterns are composed of is over 1300.

3.3 Validation Case Study

To assess whether the pattern catalogue is applicable in typical activities related to assurance case development and maintenance, we conducted a validation case study. Introducing patterns as part of maintenance activities (updates and modifications applied to already developed assurance case) is more demanding than using patterns during the development. In development, the author introduces an "empty" pattern while building a part of assurance case and instantiates it (fills with system-specific content). In maintenance, additionally a part of assurance case has to be restructured, some arguments modified, new elements added, while others relocated – and all that without losing any existing essential information and in such way that the resulting overall assurance case is valid and convincing.

We decided to use an existing assurance case dedicated to safety and operability of medical devices, developed by Kansas State University as part of the project commissioned by U.S. Food and Drug Administration (FDA) [28]. It was not a case for a specific product. Its purpose was to be a generic assurance case for a Patient Control Analgesia (PCA) pump serving as an example for infusion pump manufacturers, who, according to recent guidelines [4], are expected to deliver to FDA assurance cases for their products. This case was developed using NOR-STA and is freely available at [29].



- i8: A security case pattern
  - General Information
    - Author(s): John Goodenough, Howard F. Lipson, and Charles B. Weinstock
    - Created: 04/01/2007
    - Last modified: 04/11/2014
    - Source: Goodenough J., Lipson F. H., Weinstock Ch. B., Arguing Security - Creating Security Assurance Cases
  - Structure
    - Security: The system is acceptably secure
      - Context: Acceptably secure for this system is defined in requirements document
      - Address the security deficiencies that can be introduced in the different stages of the software development life cycle
        - Rationale
        - Requirements Deficiencies
          - Undeveloped Entity
        - Design Deficiencies
          - Undeveloped Entity
        - Coding Defects
          - Prevention/Detection
            - Rationale by: claim and argumentation strategy
            - Coding defects addressed
              - Coding Defects
                - Rationale by claim
                - There are no [Coding Defect Type X] possibilities in the code
            - Programmers Trained
              - Context: Coding Practices
              - Undeveloped Entity
        - Operational Deficiencies
          - Undeveloped Entity
  - Additional information
    - Description
    - Known uses

Fig. 2. Security Case Pattern represented in NOR-STA tool.

Being a generic example, the case was not complete e.g. several lower-level claims depending on device's design decisions remained undeveloped. The case reflected FDA's requirements e.g. to address all hazard categories (electrical, software, bio-chemical etc.), to show that remaining risk after mitigation of all hazard can be considered acceptable etc. The resulting argument was quite large as it included about 750 elements. No patterns were explicitly used in it. In the case study:

- On the methodological level, we intended to verify whether the patterns from our catalogue are applicable for a maintenance of a non-trivial assurance case;
- On the operational level, we wanted to find out how easy/difficult can such operation be done using NOR-STA tool functionality.

One of us analyzed the PCA pump assurance case and identified the parts where patterns could be introduced. The results were reviewed by a second person and resulted in some change proposals. After agreement was reached, a new version of the assurance case was created by copying patterns from the catalogue, pasting them to a given part of the case and restructuring this part, so the pattern became an integral part of argument structure and was filled with specific contents. Table 2 provides a summary about patterns used and the parts of assurance case they were introduced to.

Table 1. Patterns included in the catalogue and their sources

Sources	Patterns
Safety Case Patterns Catalogue (T. Kelly) [6]	ALARP (As Low as Reasonably Practicable) Argument
	Hazard Directed Integrity Level Argument
	Control System Architecture Breakdown Argument
	Diverse Argument
	Safety Margin
	Fault Tree Evidence
A Software Safety Pattern Catalogue (R. Weaver) [24]	Component Contributions to System Hazards
	Hazardous Software Failure Mode Decomposition
	Hazardous Software Failure Mode Classification
	Software Argument Approach
	Absence of Omission Hazardous Software Failure Mode
	Absence of Commission Hazardous Software Failure Mode
	Absence of Early Hazardous Software Failure Mode
	Absence of Late Hazardous Software Failure Mode
	Absence of Value Hazardous Software Failure Mode
	Handling of Hardware/Other Component Failure Mode
	Effects of Other Components
	Handling of Software Failure Mode
The Software Safety Argument Pattern Catalogue (R. Hawkins, T. Kelly) [25]	High-Level Software Safety Argument Pattern
	Software Contribution Safety Argument Pattern
	SSR Identification Software Safety Argument Pattern
	Hazardous Contribution Software Safety Argument Pattern
	Software Contribution Safety Arg. Pattern with Grouping
Safety Cases for Advanced Control Software: Safety Case Patterns (J. McDermid, R. Alexander, T. Kelly, Z. Kurd) [22]	Improved Safety Argument
	Maintained Safety Argument
	At Least As Safe Argument
	Risk Acceptance Argument
	Top Level System-to-Software Hazard Mitigation Argument
	Top Level System-to-Software Hazard Contribution Arg.
	Software Hazard Contributions Argument
	Hazardous Software Failure Mode Acceptability Argument
	Hazardous Software Failure Mode Absence Argument
	Safe Adaptation Argument
	Behavioural vs. Model-Building Adaptation Argument

(*continued*)

Table 1. (*continued*)

Sources	Patterns
COTS Safety Patterns (F. Ye) [19]	COTS Component Use Safety Argument
	Process-Based COTS Safety Argument
Decomposition Patterns (S. Yamamoto) [23] – conference presentation slides	Architecture Decomposition
	Functional Decomposition
	Attribute Decomposition
	Infinite Set Decomposition
	Complete Decomposition
	Monotonic Decomposition
	Decomposition by Concretion
Arguing Security (Weinstock et al.) [26]	A Security Case Pattern
Model-Based Develop-ment (Ayoub et al.) [20]	From_to Pattern

Table 2. Case study summary

Assurance case part	Pattern introduced
Arguing system safety by addressing pre-defined categories of hazards	Component Contributions to System Hazards [24]
Mitigation of "Incorrect flow rate" hazard by providing built-in alarms	Monotonic Decomposition [23]
Arguing PCA pump performs intended function on the basis or valid specification and correct implementation	from_to [20]

Several more patterns could be used, but they would affect the same parts as those listed in Table 2, therefore it was a choice between alternatives (e.g. Component Contributions to System Hazards [24] and Architectural Decomposition [23]). Introduction of patterns improved the assurance case, for example "from_to" pattern required adding definitions of intended use and intended environment, which were not explicitly expressed in the original case.

4 Conclusions and Further Work

The work reported in previous sections resulted in a catalogue of assurance case patterns grouping 45 patterns gathered from available literature. Of course, more patterns could be included, however it was our explicit decision to be selective and take into consideration only the universal, not domain-specific ones. The catalogue is the end result of an extensive literature overview. Together with our working materials and reference lists it can be considered a snapshot of assurance case patterns research & practice state in a given moment of time. Our catalogue is available to any Internet user.

This also serves as a way to disseminate knowledge about existing patterns. The registered NOR-STA users can utilize it by copying and manually instantiating patterns of their choice in their own assurance cases. The feasibility of such operations was validated in the performed case study.

In future, we plan to introduce automated pattern instantiation in NOR-STA tool. Basically, it means implementing software tool functionality which allows user to select a pattern and a source of data necessary to fill the contents of such pattern and then the instantiation is done by the tool. For example, it could be a pattern related to hazards (like ALARP Pattern [6]) and an external file storing hazard analysis results in a specified format. It is currently a subject of active research at Gdańsk University of Technology, which already resulted in elaborating automated instantiation method and first working software prototype [30]. The catalogue can be easily extended with additional patterns and we intend to do it as new patterns appear in the literature. Also, domain-specific patterns, which were rejected during catalogue development process, can be added in future (if for example there is such demand from NOR-STA users) or included in separate, domain-specific catalogues.

References

1. Kissel, R.: Glossary of key information security terms. Revision 2, NIST IR 7298. National Institute of Standards and Technology (2013)
2. International Organization for Standardization (ISO): ISO/DIS 26262: Road Vehicles - Functional Safety (2011)
3. CENELEC: EN 50126. Railway Applications: The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) (1999)
4. FDA: Infusion Pumps Total Product Life Cycle, Guidance for Industry and FDA staff (2014)
5. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)
6. Kelly, T.: Arguing safety – a systematic approach to safety case management. Ph.D. thesis, Department of Computer Science, University of York (1998)
7. Maguire, R.: Safety Cases and Safety Reports: Meaning, Motivation and Management. Ashgate Publishing Ltd, Aldershot (2006)
8. Despotou, G., Kelly, T.: Extending the safety case concept to address dependability. In: Proceedings of 22nd International System Safety Conference, pp. 645–654 (2004)
9. International Organization for Standardization (ISO): 15026-2:2011: Systems and Software Engineering – Systems and Software Assurance – Part 2: Assurance Case (2011)
10. Object Management Group: Structured Assurance Case Metamodel ver. 1.1 (2015)
11. Adelard: Claims, Arguments and Evidence (CAE). http://www.adelard.com/asce/choosing-asce/cae.html
12. GSN Community Standard Working Group: GSN community standard version 1 (2011). http://www.goalstructuringnotation.info/
13. Argevide: NOR-STA Argument Notation White Paper. https://www.argevide.com/sites/default/files/docs/Argevide%20WP2%20-%20NOR-STA%20argument%20notation.pdf
14. Górski, J., Jarzębowicz, A., Leszczyna, R., Miler, J., Olszewski, M.: An approach to trust case development. In: Proceedings of the 22nd International Conference on Computer Safety, Reliability and Security (SAFECOMP 2003). LNCS, vol. 2788, pp. 193–206 (2003)

15. Kelly, T., McDermid, J.: Safety case construction and reuse using patterns. In: Proceedings of SAFECOMP 1997, pp. 55–69 (1997)
16. Hawkins, R., Clegg, K., Alexander, R., Kelly, T.: Using a software safety argument pattern catalogue - two case studies. In: Proceedings of the 30th International Conference on Computer Safety, Reliability and Security (SAFECOMP 2011). LNCS, vol. 6894, pp. 185–198 (2011)
17. Khalil, M., Schätz, B., Voss, S.: A Pattern-based approach towards modular safety analysis and argumentation. In: Proceedings of ERTS 2014, Toulouse, France. LNCS, vol. 8822, pp. 137–151 (2014)
18. Hauge, A., Stølen, K.: A pattern-based method for safe control systems exemplified within nuclear power production. In: Proceedings of the 31st International Conference on Computer Safety, Reliability and Security (SAFECOMP 2012). LNCS, vol. 7612, pp. 13–24 (2012)
19. Ye, F.: Justifying the use of COTS components within safety critical applications. Ph.D. thesis, Department of Computer Science, University of York (2005)
20. Ayoub, A., Kim, B., Lee, I., Sokolsky, O.: A safety case pattern for model-based development approach. In: Proceedings of the 4th NASA Formal Methods Symposium (NFM 2012). LNCS, vol. 7226, pp. 141–146 (2012)
21. Denney, E., Pai, G.: Safety case patterns: theory and applications. NASA/TM–2015–218492 Technical report (2015)
22. Alexander, R., Kelly, T., Kurd, Z., McDermid, J.: Safety cases for advanced control software: safety case patterns. Technical report, University of York (2007)
23. Yamamoto, S., Matsuno, Y.: An evaluation of argument patterns to reduce pitfalls of applying assurance case. In: Proceedings of 1st International Workshop on Assurance Cases for Software-Intensive Systems (ASSURE 2013), pp. 12–17 (2013)
24. Weaver, R.: The safety of software – constructing and assuring arguments. Ph.D. thesis, Department of Computer Science, University of York (2003)
25. Hawkins, R., Kelly, T.: A software safety argument pattern catalogue. Technical report, University of York (2013)
26. Weinstock, C., Lipson, H., Goodenough, J.: Arguing security - creating security assurance cases. US CERT BSI (Build Security In) report, Carnegie Mellon University (2007)
27. Assurance Case Patterns On-line Catalogue, Gdańsk University of Technology. http://www.nor-sta.eu/en/en/news/assurance_case_pattern_catalogue
28. Larson, B.R., Hatcliff, J., Chalin, P.: Open source patient-controlled analgesic pump requirements documentation. In: 5th International Workshop on Software Engineering in Health Care (SEHC), pp. 28–34 (2013)
29. Larson, B.R.: Open PCA Pump Assurance Case, Santos Research Group, Kansas State University (2014). http://openpcapump.santoslab.org/
30. Wardziński, A., Jarzębowicz, A.: Towards safety case integration with hazard analysis for medical devices. In: Proceedings of 4th International Workshop on Assurance Cases for Software-intensive Systems (ASSURE 2016). LNCS, vol. 9923, pp. 87–98 (2016)

Information System as a Cause of Cargo Handling Process Disruption in Intermodal Terminal

Justyna Świeboda and Mateusz Zając(✉)

Wroclaw University of Technology,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
{justyna.swieboda,mateusz.zajac}@pwr.edu.pl

Abstract. The issue of reliability in the cargo handling process is very important. The particular importance is the level of cargo service in relation to intermodal transport. The specificity of this way of good transportation require high effectiveness in the mutual interaction of information systems, technology and man activity. Container handling process on intermodal terminal is presented and divided into five pub-processes. The aim of the article is to present level of disruptions coming from information fail. The Authors focused on information system that appear the weakest among the primary reasons of disruptions. The article includes errors selection due to the cause of their formation and their impact on the correctness of the process implementation.

Keywords: Intermodal transport · Intermodal terminal · Information flow

1 Introduction

Intermodal transport [3] is defined as passenger or freight transport from the initial point to the destination, using at least two means of transport. Regarding freight transport such process can be carried out with a various configuration of the means of transport, i.e. by truck, rail and ocean shipping. Intermodal transport refers to a multimodal chain of freight service; most frequently this is an integrated freight unit, namely a container. Such transport is carried out routes usually over the distance of 300 km.

Intermodal terminal infrastructure is essential to the implementation of the intermodal transport. Part of the point infrastructure includes seaports, inland terminals and depots (place of inspection, storage and repair empty containers) [26]. Wide information on cargo operation in the terminals can be find in [19].

There are two flows in inland terminal, as in average logistics system: flow of cargo and information. Elements of the system are [1, 12]:

- the technical means that allow loading, unloading, handling among the storage area,
- integrated computer networks for the information transmission between users of the system, as well as to improve the work of the system,
- experts, who are responsible for taking the decision, as well as commanding the implementation of scheduled tasks.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_40

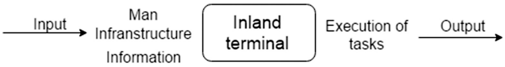


Fig. 1. An inland terminal as system.

The inputs to container handling system on the terminals (Fig. 1) are: man activity, infrastructure and information. The output is tasks execution. Container terminals, constitute a key role in intermodal transport, therefore it is important that the terminals worked in dependable way.

Currently logistics systems are highly developed, that the search for reasons of disruptions only in technical devices is insufficient. Disruptions of logistics systems and logistics processes should be analyzed from the human activity point of view and the information sharing becomes significant reasons of insufficient level of service.

The article presents the cargo handling process model regarding in intermodal terminal operations. The contribution includes as identification and the determination of the disruption causes. The Authors focused on information system. This element appears the weakest among the primary reasons of disruptions.

2 Literature Review

Dependability is defined as the ability of an object to fulfil the present requirements [7]. In this context dependability can be meant as an ability to provide the process of cargo flow at intermodal transport. The issue of reliability has been brought up in many studies concerning various systems or technical facilities. Reliability, susceptibility, resistance in a logistics system has been discussed in different studies [13, 21, 22, 24, 25, 28]. Dependability is also described in transport cases; in railway transport [17, 18] or air transport [8–10, 20]. Dependability modeling and analysis of technical objects and systems is described in e.g. [11, 23, 24]. Dependability model of intermodal terminals is presented in [27]. An essential element in the system, whereby handling operations are carried out, is also given as dependability of the information system as presented in the paper [14, 15]. In those papers system oriented analysis is presented, however there is lack of process oriented studies in literature.

As previously mentioned in any logistic system includes the flow of cargo and accompanying flow of information. Information, is considered as an element of the information system. Mostly, the systems are examined in terms of security, secure data against theft or data recovery. For example in [5] effectiveness of information system is identified as relationship between: system quality, information quality, user, user satisfaction, individual impact and organizational impact. In paper [16] (on the basis of method SERVQUAL) are distinguished some aspects, which have influence on service quality: tangibles, reliability, responsiveness, assurance, empathy.

The next group is information sharing between users of supply chain. At this group, many authors are considered various cases, for example: complete information sharing, partial information sharing or no information sharing between different users in supply chain. For example works [2, 4] proposed quantitative modeling of information - value of information, in supply chain – by Markov model. Information value was obtained to identify advantage coming from information sharing between supplier and retailer. In paper [6] are considered three models: in the first one was no information sharing, in the second was sharing partial information from retailer to supplier, and the last one was complete information sharing between all users in supply chain. In those paper authors are used Markov process.

In the available literature the authors analyze the information system, not the process. The authors assess the information system in the selected organization (usually enterprises) as a whole, do not consider single message. In the present work, at any level of the process, the information were extracted, considered specification what the data should contain. Basis on this information research and evaluation of logistics process was made.

In the further part of the contribution results of a study on intermodal freight service on terminal are presented. The research includes causes of disruption of the logistics process (container cargo handling service) consisting of five stages – sub processes.

3 The Dependability Model of Intermodal Cargo Service

The process of container service in intermodal terminal is presented on Fig. 2. It begins by sending the customer's order to the company, and ends at the time of cargo delivery to the client.

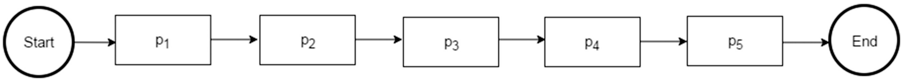


Fig. 2. The process of cargo in intermodal.

Figure 3 presents particular processes of container cargo handling:

- p_1 – the process of preparation of the order by the customer,
- p_2 – order input to a computer system,
- p_3 – acceptance of cargo in the container terminal, integrated loading units (perhaps using the tractor or train),
- p_4 – cargo operations; vehicle/train/transshipment operations in order to service the goods on terminal,
- p_5 – cargo units transport to the customer.

It has been extracted the flow of information and has been highlighted performance of different actions and operations. The graph of information flow model is shown on Fig. 3. In this contribution information is defined as a data notification or communication equated with the message [2].

In accordance with the cargo handling process there are 5 operations assigned as $(e)_{i;}$ the performance of the tasks is marked as r_i and 4 flows of information marked as i_i. The cargo handling process begins with the preparation of the order (*p1* - according to Fig. 2) by the customer and are willing tasks determined by the physical send the information from the client to the freight forwarder at the terminal land (e_1). Freight forwarder then enters the data delivery to the information system (p_2) and sends this

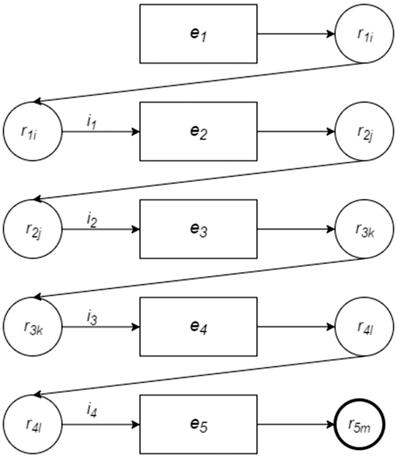


Fig. 3. Model the process of cargo handling in intermodal terminal

request to the machine operators (e_2). The next process is the adoption of the integrated loading units on the terminal, and then there is the physical check on the condition of the containers (or not damaged), and to verify compliance of delivery (p_3), on the basis of the freight forwarder assumes the transport order for cargo (e_3) (deciding whether the cargo is to be stored on the terminal or directly sent to the client). After the completion of this task is the process of moving cargo (p_4), followed by translocation operations (e_4), to them it is loading, unloading and storage units. The last process (p_5) is the loading of the means of transport and the means of transport from terminal to the client (e_5). As can be seen from Fig. 3, each task execution (e_i) is preceded by the receipt of information (i_i) about its execution. The first message (i_1) contains data about cargo, which is to be delivered to the terminal, e.g. the number of cargo, weight of cargo, ADR/RID, etc. The next information (i_2) applies to data delivery, which is what (the number of the container) cargo will be delivered, the day of delivery, and what means of transport is to be involved. The next information (i_3) is created when you create the order in the system by the freight forwarder, or indicate which cargo is to be stored, and that cargo is to be sent and what day. Most recent information (i_4) is basically a waybill, which is passed to the person who performs the transport to the customer with a load. The last element in the model presented in Fig. 3 is the realization of tasks. In the case of the implementation of the tasks we have to do with the r_1, r_2, r_3, r_4, r_5, and this realization has an index i, j, k, l, m <1, 2, 3>, this means that the implementation of tasks can be performed in 3 different ways. As 1 marked the full (complete) implementation tasks, as 2 marked a partial implementation of the tasks and as 3 adopted the lack of execution of the task. For example, the r_{32} will mean partly implementation of e_3, while the r_{43} will mean the lack of implementation of tasks of the e_4 and to stop the process at this point. One of the assumptions of this model is that each next implementation task depends on the result of previous which means that to be possible departure of the cargo from the terminal, must be made the previous task, at least at the level of partial implementation.

Regarding the model presented on Fig. 3, Sect. 4 consist of the distortion analysis of the cargo service process. The results of each sub-process is described, specifically their validity with respect to the smooth process of cargo handling in intermodal terminal.

4 Characteristics of Cargo Handling Process Disturbances in Intermodal Terminal

In 2016, the study associated with the level of customer service, in accordance with the cargo handling process model (Fig. 2) were carried out in one of polish inland container terminals. The observation has been concerned on the cargo flow from the railway track at the entrance to the terminal to the cargo delivery to the customer by road. Total number of observed services counted 948 numbers.

It was assumed, that each of the five stages may end in one of three ways:

1. full compliance with the expectation of the client (internal or external): compliance with the timetable, the time compliance specification, the process is entirely correctly; further marked as G;
2. partial compliance: there may be a small documentation non-compliance (e.g. the lack of complementary document, time discrepancies up to 2 h); the process is disrupted, however, may end up successfully; the fulfillment of the client expectations; further marked as S;
3. non-compliance process: the lack of timely delivery of cargo, lack of required documents, etc., marked as F.

The distortion created in the course of handling cargo are classified into 1 of 5 groups:

men, technique, information, previous delays, other.

In the further part the nature of each of the causes of disruption is showed.

Men – the cause of the disruption is the man; the lack of a suitable person for the position, the lack of a decisive person or contractor, human error, the disruption clearly depends on the intentional or non-intentional human activities or lack of appropriate response;

Technique – the cause of the disruption is the unreliability of a technical device or machine; in the case of electronic devices is a lack of communication, lack of services availability, the update of the operating system at the time required the use of the equipment; in the case of transport equipment-lack of availability, failure, repair, preventive maintenance, etc.

Information – the lack of satisfactory information on time, in an appropriate form, the lack of a clear message, the information too late, etc.

Previous delays – delay of service resulting from non-compliance on previous stages; the cascading delays,

Other – delay incoming not from technology, management or operation processes, e.g. causes of weather, external reasons, traffics, ect.

The Fig. 4 shows the structure of the achieved levels of validation the cargo flows process at the following stages of the operation. As it can be seen, along with the process progress the effectiveness of the correct implementation decreases. In the first stage, there were a total of 213 cases of non-compliance. Later, this number increasing, with the exception of phase 3, where the total number of non-compliance decreased from 254 (step 2) to 246 (step 3). However, the total number of processes that are able to take increased then from 109 to 125.

Figure 5 shows the causes of disruptions. As it can be seen many inconsistencies in the handling of cargo was apparent from the terms of service are not met in the previous stage; for the most part the reason were considerable delays. Cascade delays are the indirect cause of disruption. Looking at the root causes (direct causes) of

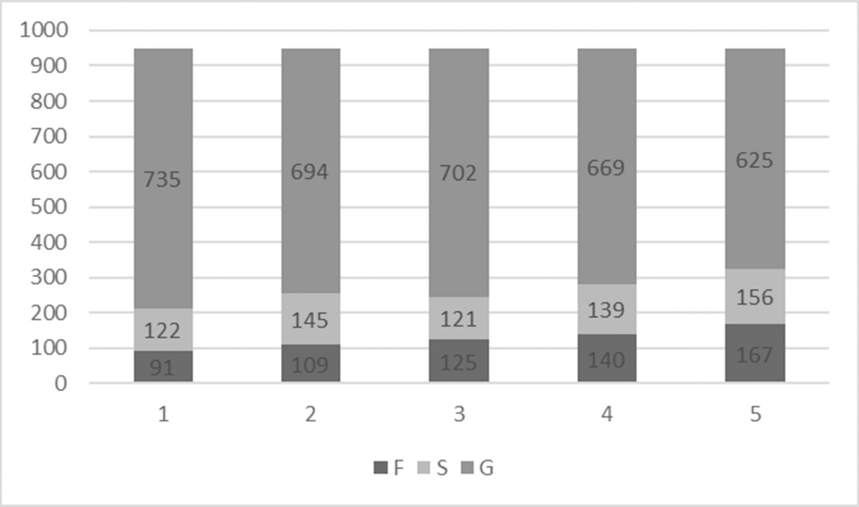


Fig. 4. The correctness of the process at the stages

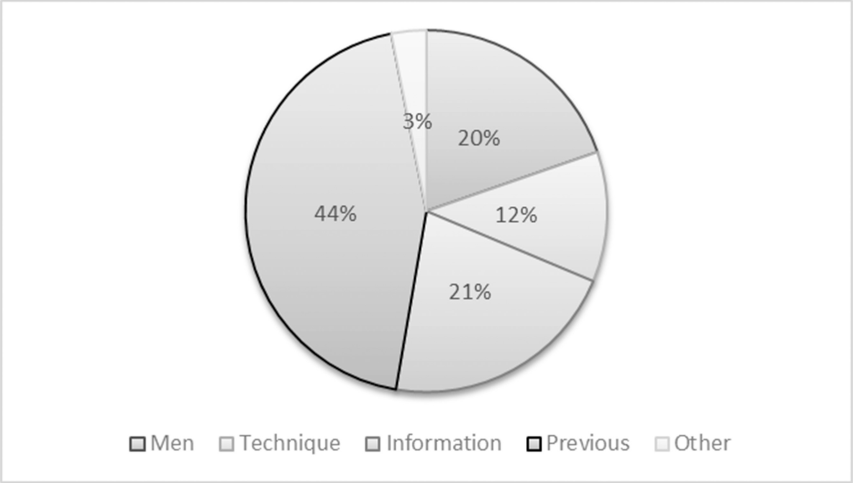


Fig. 5. Disruptions, including direct and indirect causes

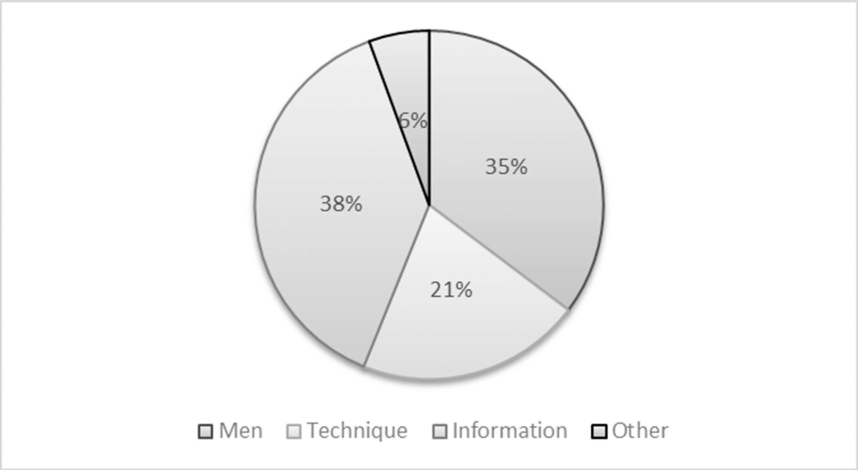


Fig. 6. Disruptions, including only direct causes

non-compliance, it appears that the most common cause of disruption is wrong, inadequate information system. Next, man activity and lack of equipment (technique) provide disruptions. The information about roots presented on Fig. 6.

The structure of disruptions changes in successive stages. Figure 7 shows the direct and indirect distortion, and in Fig. 8 direct cases only. Indirect disruptions are clearly dominant. The average value of cascade disturbances is 44.18%, however, in the first stage it was 18.77%, while in fifth 56.96%. The last two stages are dominated by the indirect disruption, in both count more than 50% of all disturbances of the process. The direct distortion are characterizing by the variability. Depending on the stage, different

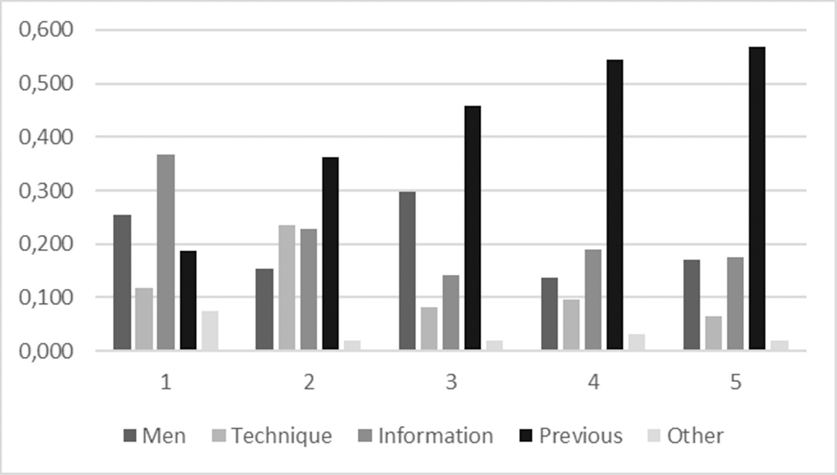


Fig. 7. The structure of the disturbance process in stages, direct and indirect causes

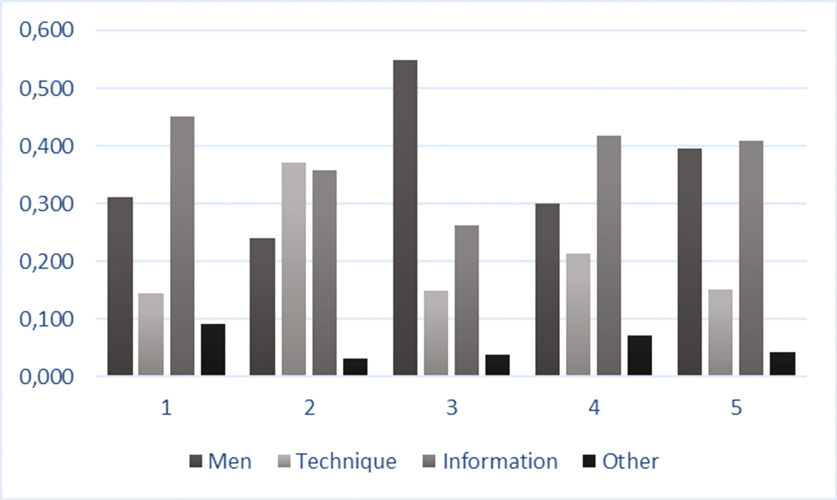


Fig. 8. The structure of the disturbance process in stages, direct causes

disturbances seem to be dominant. As far as the first stage of information (lack) is the most dangerous distorting cause, in the third stage clearly dominates the human factor, as the reason for the lack of conformity process. In the successive stages of the disruption structure changes are visible.

5 Summary

The problem of reliability in the cargo handling operations is a very important topic, due to the customer service do not meet the customer expectations. The expectations are closely linked with the searching for factors that allow for fluent logistic service. Disruptions in the cargo handling can be divided into direct and indirect. The direct factors, in the case of container handling operation in intermodal terminal, have been divided into those that derive from information, human and technology errors. As shown in the publication, in this nowadays technique/technology is not the primary problem in meeting customer expectations. Today, many interference results from direct human activity, inefficient, poorly structured system information.

In most cases, the number of interference increases with successive stages of the process. There has been a decrease in the number of incorrect services for transition from the sub-process 2 to 3. In a sense it can be understand as an activity characterised by resistance to interference, especially when number of cascading delays decreased.

Further work will rely on the detailed examining the relationships between information flow and loading operations. In the context of supply chain reliability it seems to be important to examine completeness and clarity of information, consequently the response in the form of the effective implementation of the process at the operational level.

References

1. Abt, S.: Systemy logistyczne w gospodarowaniu, Teoria i praktyka logistyki, Wyd. Akademii, Ekonomicznej w Poznaniu, Poznań (1996)
2. Bazewicz, M.: Wstęp do systemów informatycznych i reprezentacji wiedzy. Wydawnictwo Politechniki Wrocławskiej, Wrocław (1993)
3. Crainic, T.G., Kim, K.H.: Intermodal transportation. Transportation **14**, 467–537 (2006)
4. Davis, L.B., King, R.E., Hodgson, T.J., Wei, W.: Information sharing in capacity constrained supply chains under lost sales. Int. J. Prod. Res. **49**(24), 7469–7491 (2011)
5. DeLone, W.H., McLean, E.R.: Information systems success: the quest for the dependent variable. Inf. Syst. Res. **3**(1), 60–95 (1992)
6. Helper, C.M., Davis, L.B., Wei, W.: Impact of demand correlation and information sharing in a capacity constrained supply chain with multiple-retailers. Comput. Ind. Eng. **59**(4), 552–560 (2010)
7. IEC 60050-192:2015, International electrotechnical vocabulary - Part 192: Dependability
8. Kierzkowski, A.: Method for management of an airport security control system. Proc. Inst. Civ. Eng. Trans. (2016). http://dx.doi.org/10.1680/jtran.16.00036
9. Kierzkowski, A., Kisiel, T.: Simulation model of security control system functioning: a case study of the Wroclaw Airport terminal. J. Air Trans. Manag. (2016). http://dx.doi.org/10.1016/j.jairtraman.2016.09.008
10. Kierzkowski, A., Kisiel, T.: A model of check-in system management to reduce the security checkpoint variability. Simul. Model. Pract. Theory **74**, 80–98 (2017). http://dx.doi.org/10.1016/j.simpat.2017.03.002
11. Kisiel, T., Valis, D., Zak, L.: Application of regression function - two areas for technical system operation assessment. In: Proceedings of CLC 2013: Carpathian Logistics Congress, TANGER Ltd., pp. 500–505 (2014)
12. Nowakowski, T.: Niezawodność Systemów Logistycznych. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2011)
13. Nowakowski, T., Werbinska-Wojciechowska, S.: Data gathering problem in decision support system for means of transport maintenance processes performance development. In: Safety, Reliability and Risk Analysis: Beyond the Horizon, Proceedings of 22nd Annual Conference on European Safety and Reliability (ESREL) 2013, pp. 899–907 (2014)
14. Polak, R., Laskowski, D.: Reliability of routing protocols. J. KONBiN **35**(1), 51–62 (2015)
15. Polak, R., Laskowski, D.: Network reliability with use of various transmission media. J. KONBiN **39**(1), 57–78 (2016)
16. Pitt, L.F., Watson, R.T., Kavan, C.B.: Service quality: a measure of information systems effectiveness. MIS Q. **19**, 173–187 (1995)
17. Restel, F.J.: The Markov reliability and safety model of the railway transportation system. In: Safety and Reliability: Methodology and Applications - Proceedings of the European Safety and Reliability Conference, ESREL 2014 (2015)
18. Restel, F.J., Zajac, M.: Reliability model of the railway transportation system with respect to hazard states. In: IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 1031–1036 (2015). doi:10.1109/IEEM.2015.7385805
19. Rožić, T., Rogić, K., Bajor, I.: Research trends of inland terminals: a literature review. Promet-Traffic Transp. **28**(5), 539–548 (2016)
20. Siergiejczyk, M., Krzykowska, K., Rosiński, A.: Reliability assessment of cooperation and replacement of surveillance systems in air traffic. In: Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, pp. 403–411. Springer (2014)

21. Świeboda, J., Zając, M.: Studies on information subsystem operation in container terminal based on simple example. In: Risk, Reliability and Safety: Innovating Theory and Practice: Proceedings of ESREL 2016, pp. 2023–2030 (2016)
22. Tubis, A., Werbińska-Wojciechowska, S.: Inventory management of operational materials in road passenger transportation company - case study. In: CLC 2013: Carpathian Logistics Congress - Congress Proceedings, pp. 65–70 (2014)
23. Vintr, Z., Valis, D.: A tool for decision making in K-out-of-N system maintenance. In: Applied Mechanics and Materials, vol. 110, pp. 5257–5264. Trans Tech Publications (2012). doi:10.4028/www.scientific.net/AMM.110-116.5257
24. Vintr, Z., Valis, D.: Vehicle maintenance process optimization using life cycle costs data and reliability-centered maintenance. In: Proceedings of the First International Conference on Maintenance Engineering, pp. 180–188 (2006). ISBN 978-7-03-018064-3
25. Woźniak, W., Wojnarowski, T.: A method for the rapid selection of profitable transport offers within the freight exchange market. In: 25th IBIMA Conference, pp. 2073–2085 (2015)
26. Zajac, M., Swieboda, J.: An unloading work model at an intermodal terminal. In: Theory and Engineering of Complex Systems and Dependability, pp. 573–582. Springer (2015). doi:10.1007/978-3-319-19216-1_55
27. Zajac, M., Swieboda, J.: The method of error elimination in the process of container handling. In: 2015 International Conference on Military Technologies (ICMT), pp. 1–6. IEEE (2015). doi:10.1109/MILTECHS.2015.7153699
28. Zając, P.: Transport-storage system optimization in terms of exergy. In: Proceedings of the 13th International Conference on Harbor Maritime Multimodal Logistics Modeling & Simulation, HMS, pp. 143–148 (2010). ISBN 978-2-9524747-4-0

Anticipation Scheduling in Grid Virtual Organizations

Victor Toporkov(✉), Dmitry Yemelyanov, Vadim Loginov, and Petr Potekhin

National Research University “MPEI”, ul. Krasnokazarmennaya, 14,
Moscow 111250, Russia
{ToporkovVV, YemelyanovDM, LoginovVA, PotekhinPA}@mpei.ru

Abstract. In this work, a job-flow scheduling approach for Grid virtual organizations is proposed and studied. Users’ and resource providers’ preferences, virtual organization’s internal policies, resources geographical distribution along with local private utilization impose specific requirements for efficient scheduling according to different, usually contradictive, criteria. With increasing resources utilization level the available resources set and corresponding decision space are reduced. This further complicates the task of efficient scheduling. In order to improve overall scheduling efficiency we propose a heuristic anticipation scheduling approach. It generates a near optimal but infeasible scheduling solution and includes special replication procedure for efficient and feasible resources allocation.

Keywords: Scheduling · Grid · Resources · Utilization · Heuristic · Job batch · Virtual organization · Anticipation · Replication

1 Introduction and Related Works

In distributed environments with non-dedicated resources such as utility Grids the computational nodes are usually partly utilized by local high-priority jobs coming from resource owners. Thus, the resources available for use are represented with a set time intervals (slots) during which the individual computational nodes are capable to execute parts of independent users’ parallel jobs. These slots generally have different start and finish times and a performance difference. The presence of a set of slots impedes the problem of resources allocation necessary to execute the job flow from computational environment users. Resource fragmentation also results in a decrease of the total computing environment utilization level [1, 2].

Application level scheduling [3] is based on the available resources utilization and, as a rule, does not imply any global resource sharing or allocation policy. Job flow scheduling [4, 5] in user’s virtual organizations (VO) suppose uniform rules of resource sharing and consumption, in particular based on economic models. This approach allows improving the job-flow level scheduling and resource distribution efficiency.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_41

VO policy may offer optimized scheduling to satisfy both users' and VO common preferences. The VO scheduling problems may be formulated as follows: to optimize users' criteria or utility function for selected jobs [6, 7], to keep resource overall load balance [8, 9], to have job run in strict order or maintain job priorities [10], to optimize overall scheduling performance by some custom criteria [11, 12], etc.

VO formation and performance largely depends on mutually beneficial collaboration between all the related stakeholders. However, users' preferences and owners' and administrators' preferences may conflict with each other. Users are likely to be interested in the fastest possible running time for their jobs with least possible costs whereas VO preferences are usually directed to available resources load balancing or node owners' profit boosting. Thus, VO policies in general should respect all members and the most important aspect of rules suggested by VO is their fairness. A number of works understand fairness as it is defined in the theory of cooperative games, such as fair job flow distribution [9], fair quotas [13, 14], fair user jobs prioritization [10], and non-monetary distribution [15]. The cyclic scheduling scheme (CSS) [16] implements a fair scheduling optimization mechanism which ensures stakeholders interests to some predefined extent. The downside of a majority centralized metascheduling approaches is that they lose their efficiency and optimization features in distributed environments with a limited resources supply. For example in [2], a traditional backfilling algorithm provided better scheduling outcome when compared to different optimization approaches in resource domain with a minimal performance configuration. The general root cause is that in fact the same scarce set of resources (being efficient or not) have to be used for a job flow execution or otherwise some jobs might hang in the queue. Under such conditions, user jobs priority and ordering greatly influence the scheduling results. At the same time, application-level brokers are still able to ensure user preferences and optimize the job's performance under free-market mechanisms.

Main contribution of this paper is a heuristic CSS-based job-flow scheduling approach which retains efficiency even in distributed computing environments with limited resources. Special scheduling solution *replication* procedure is proposed and studied to ensure a feasible scheduling result. The rest of the paper is organized as follows. Section 2 presents a general CSS fair scheduling concept. The proposed heuristic-based scheduling technique is presented in Sect. 3. Section 4 contains simulation experiment setup and results for the proposed scheduling approach. Finally, Sect. 5 summarizes the paper.

2 Cyclic Alternative-Based Fair Scheduling

Scheduling of a job flow using CSS is performed in time cycles known as scheduling intervals, by job batches [16]. The actual scheduling procedure consists of two main steps. The first step involves a search for alternative scenarios of each job execution, or simply alternatives [17]. During the second step the dynamic programming methods [16] are used to choose an optimal alternatives' combination. One alternative is selected for each job with respect to the given VO and user criteria. An example for a user scheduling criterion may be an overall job running time, an overall running cost, etc. This criterion describes user's preferences for that specific job execution and expresses a type of an additional optimization to perform when searching for alternatives. Alongside with time (*T*) and cost (*C*) properties each job execution alternative has a user utility (*U*) value: user evaluation against the scheduling criterion. A common VO optimization problem may be stated as either minimization or maximization of one of the properties, having other fixed or limited, or involve Pareto-optimal strategy search involving both kinds of properties [4, 16, 18]. For a fair CSS scheduling model the second step VO optimization problem could be in form of: $C \rightarrow$ max, lim U (maximize total job flow execution cost, while respecting user's preferences to some extent); $U \rightarrow$ min, lim T (meet user's best interests, while ensuring some acceptable job flow execution time) and so on [16].

We consider the following relative approach to represent a user utility U. A job alternative with the minimum (best) user-defined criterion value Z_{min} corresponds to the left interval boundary ($U = 0\%$) of all possible job scheduling outcomes. An alternative with the worst possible criterion value Z_{max} corresponds to the right interval boundary ($U = 100\%$). In the general case, for each alternative with value Z, U is set depending on its position in [Z_{min}; Z_{max}] interval using the following formula: $U = \frac{Z - Z_{\text{min}}}{Z_{\text{max}} - Z_{\text{min}}} * 100\%$. Thus, each alternative gets its utility in relation to the "best" and the "worst" optimization criterion values user could expect according to the job's priority. And the more some alternative corresponds to user's preferences the smaller is the value of U. For a fair scheduling model the second step VO optimization problem could be in form of: $C \rightarrow$ max, lim U (maximize total job flow execution cost, while respecting user's preferences to some extent); $U \rightarrow$ min, lim T (meet user's best interests, while ensuring some acceptable job flow execution time) and so on [16].

The launch of any job requires a co-allocation of a specified number of slots, as well as in the classic backfilling variation. A single slot is a time span that can be assigned to run a part of a parallel job. The target is to scan a list of N_s available slots and to select a *window* of m parallel slots with a length of the required resource reservation time. The user job requirements are arranged into a resource request containing a resource reservation time, characteristics of computational nodes (clock speed, RAM volume, disk space, operating system etc.), limitation on the selected window maximum cost. ALP, AMP and AEP window search algorithms were discussed in [17]. The job batch scheduling performs consecutive allocation of a multiple *nonintersecting* in terms of slots alternatives for each job. Otherwise irresolvable collisions for resources may occur if different jobs will share the same time-slots. Sequential alternatives search and resources reservation procedures help to prevent such scenario. However in an extreme case when resources are limited or overutilized only at most one alternative execution could be reserved for each job. In this case alternatives-based scheduling result will be no different from First Fit resources allocation procedure [2]. First Fit resource selection algorithms [19] assign any job to the first set of slots matching the resource request conditions without any optimization.

3 Anticipation Scheduling

In order to address this problem the following heuristic job batch scheduling scheme is proposed which consists of three main steps. First, a set of all possible execution alternatives is found for each job not considering time slots intersections and without any resources reservation. The resulting intersecting alternatives found for each job reflect a full range of different job execution possibilities user may expect on the current scheduling interval. Second, CSS scheduling procedure [16] is performed to select alternatives combination (one alternative for each job of the batch) optimal according to VO policy. The resulting alternatives combination most likely corresponds to an infeasible scheduling solution as possible time slots intersection will cause collisions on resources allocation stage. The main idea of this step is that obtained infeasible solution will provide some heuristic insights on how each job should be handled during the scheduling. For example, is time-biased or cost-biased execution is preferred, how it should correspond to user criterion and VO administration policy and so on. Third, a feasible resources allocation is performed by replicating alternatives selected in step 2. The base for this replication step is an Algorithm searching for Extreme Performance (AEP) described in details in [17]. In the current step AEP helps to find and reserve feasible execution alternatives most similar to those selected in the near-optimal infeasible solution. After these three steps are performed the resulting solution is both feasible and efficient as it reflects scheduling pattern obtained from a near-optimal reference solution from step 2.

We used AEP modification to allocate a diverse set of execution alternatives for each job. Originally AEP scans through a whole list of available time slots and retrieves one alternative execution satisfying user resource request and optimal according to user custom criterion. During this scan, we saved all intermediate AEP search results to a dedicated list of possible alternatives. For the replication purpose a new *Execution Similarity* criterion was introduced which helps AEP to find a window with minimum *distance* to a reference alternative. Generally, we define a *distance* between two different alternatives (windows) as a relative difference or *error* between their significant criteria values. For example if reference alternative has C_{ref} total cost, and some candidate alternative cost is C_{can}, then the relative cost error E_C is calculated as $E_C = \frac{|C_{\mathrm{ref}} - C_{\mathrm{can}}|}{C_{\mathrm{ref}}}$. If one need to consider several criteria the *distance* D between two alternatives may be calculated as a linear sum of criteria errors: $D_l = E_C + E_T + .. + E_U$, or as a geometric distance in a parameters space: $D_g = \sqrt{E_C^2 + E_T^2 + ..E_U^2}$.

AEP modification with *Execution Similarity* criterion is represented below.

```
Input Data: slotList - a list of available slots; job - a job for which the search is
performed; refAlternative – reference alternative used to find similar job execution
window.
Result: closestWindow – execution window similar to refAlternative

slotList = orderSystemSlotsByStartTime();
minDistance = MAX_VALUE;

for each slot in slotList do
        if not(properHardwareAndSoftware(job, slot.node)) then
                continue;
        end
        windowSlotList.add(slot);
        windowStartTime = slot.startTime;
        for each wSlot in windowSlotList do
                minLength = wSlot.node.getWorkingTimeEstimate();
                if (wSlot.endTime - windowStartTime) < minLength then
                        windowSlotList.remove(wSlot);
                end
        end
        if windowSlotList.size() ≥ job.nodesNeed then
        distance = calculateDistance(windowSlotList, refAlternative);
        if distance < minDistance then
                minDistance = distance;
                closestWindow = windowSlotList;
                end
        end
end
```

In this algorithm an expanded window *windowSlotList* of size *M moves* through a whole list of all available slots *slotList* sorted by their start time in ascending order. At each step any combination of *m* slots inside *windowSlotList* (in the case, when *m* *M*) can form a window that meets all the requirements to run the job. The main difference from the original AEP is that instead of searching for a window with a maximum single criterion value, we retrieve window with a minimum distance D_g or D_l to a reference execution alternative. Generally, this distance can reflect job execution preferences in terms of multiple criteria such as job execution cost, runtime, start time, finish time, etc.

For a feasible job batch resources allocation AEP consequentially allocates for each job a single execution window with a minimum *distance* to a reference corresponding alternative from an infeasible solution. Time slots allocated for *i*-th job are reserved and excluded from the slot list when AEP search algorithm is performed for the following jobs $i + 1$, $i + 2$,.. N. Thus this procedure prevents any conflicts for resources and provides scheduling solution which in some sense reflects near-optimal reference solution.

4 Simulation Study

An experiment was prepared as follows using a custom distributed environment simulator [2, 16, 17]. VO and computing environment properties:

- The resource pool includes 80 heterogeneous computational nodes.
- A specific cost of a node is an exponential function of its performance value (base cost) with an added variable margin distributed normally as ±0.6 of a base cost.
- The scheduling interval length is 800 time quanta. The initial resource load with owner jobs is distributed hyper-geometrically resulting in 5% to 10% time quanta excluded in total.

Job batch properties:

- Jobs number in a batch is 125.
- Nodes quantity needed for a job is a whole number distributed evenly on [2; 6].
- Node reservation time is a whole number distributed evenly on [100; 500].
- Job budget varies in the way that some of jobs can pay as much as 160% of base cost whereas some may require a discount.
- Every request contains a specification of a custom user criterion which is one of the following: job execution runtime or overall execution cost.

4.1 Replication Scheduling Accuracy

The first experiment is dedicated to a replication scheduling accuracy study. For this matter we conducted and collected data from more than 1000 independent job batch scheduling simulations. First, a general CSS was performed in each experiment for the following job-flow execution cost maximization problem $C \rightarrow \max$, $\lim U_a = 10\%$. U_a stands for the average user utility for one job, i.e. $\lim U_a = 10\%$ means that at average resulting deviation from the best possible outcome for each user did not exceed 10%. Next, *linear* and *geometric* replication algorithms were executed to replicate CSS solution using linear D_l and geometric D_g distance criteria. In the current experiment we used job execution cost error E_c and processor time usage error E_t to calculate distances D_l and D_g.

In order to evaluate the resulting difference in scheduling outcomes, we additionally performed CSS algorithm for $C \rightarrow \max$, $\lim U_a = 0\%$ (ensuring users' individual preferences only) and $C \rightarrow \max$, $\lim U_a = 100\%$ (ensuring VO preference, i.e. maximizing overall cost without taking into account users' criteria) problems. These additional problems reflect extreme boundaries for scheduling results, which can be used to evaluate a relative replication error. Table 1 contains scheduling results for all these three problems and two replication algorithms.

The results indicate that both linear and geometric replication algorithms provided average scheduling parameters very close to the reference solution (indicated as bold in Table 1). And especially close against job execution cost and processor time usage,

Table 1. CSS replication average scheduling results

Job execution characteristic	C -> max, lim U_a = 0%	C-> max, lim U_a = 10%	Linear replication	Geometric replication	C -> max, lim U_a = 100%
Cost	1283	***1349***	1353	1353	1475
Processor time	191.6	***191.2***	190.6	190.5	202.3
Finish time	367.1	***353.8***	356.2	356.4	358.5
U_a, %	0	***9.9***	17.6	17.8	65

i.e. characteristics which were used for a replication distance calculation. For example, *borderline* problems $C \rightarrow$ max, lim $U_a = 0\%$ and $C \rightarrow$ max, lim $U_a = 0\%$ provided average job execution cost (main job-flow optimization criterion) values 1283 and 1475 correspondingly. Reference intermediate solution provided 1349. And both replication algorithms ensured average job execution cost 1353 with only 2% deviation from reference solution against [1283; 1475] interval of possible scheduling outcomes. Although replication algorithms showed their efficiency with respect to integral job flow processing parameters (such as average job execution cost, runtime, finish time), individual user's preferences were considered to a lesser extent. It can be observed in the Table 1 that both replication algorithms provided average user utility U_a almost twice as much as the reference problem.

To address this discrepancy in more details Fig. 1 shows average linear and geometric replication distances for each job of the batch. Figure 1 shows that there values are practically independent from an ordinal job number and do not exceed 0.05. For comparison average distances between the most and the least expensive alternative executions for the first batch job amounted: $D_l = 1.15$ and $D_g = 0.88$. These values exceed average replication distances in 20 times and therefore are not shown in the Fig. 1. Thus, we can conclude that replication error for each batch job on average does not exceed 5% against interval of possible scheduling outcomes.

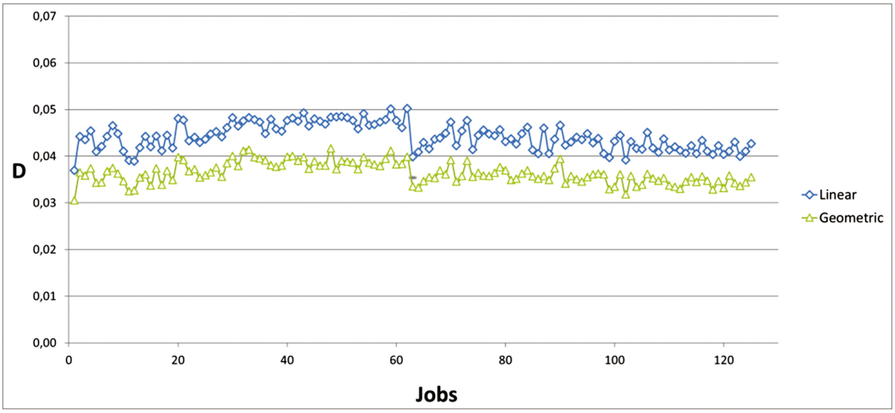


Fig. 1. Average replication error for user jobs

4.2 Anticipation Scheduling Simulation

The second experiment series consider anticipation scheduling efficiency. During each experiment a VO domain and a job batch were generated and the following scheduling schemes were simulated and studied. First, a general CSS solved the optimization problems $T \rightarrow$ min, lim U with different limits $U_a \in \{0\%, 1\%, 4\%, 10\%, 16\%, 32\%, 100\%\}$. Second, a near-optimal but infeasible reference solution REF was obtained for the same problems. Third, a replication procedure CSS_{rep} was performed based on CSS solution to demonstrate a replication process accuracy. For the heuristic anticipation scheduling ANT the same replication procedure was performed based on REF solution. We used a geometric distance as a replication criterion. Finally two independent job batch scheduling procedures were performed to find scheduling solutions most suitable for VO users (USER_{opt}) and VO administrators (VO_{opt}). USER_{opt} was obtained by using only user criteria to allocate resources for jobs without taking into account VO preferences. VO_{opt} was obtained by using one VO optimization criterion ($T \rightarrow$ min) for each job scheduling without taking into account user preferences.

1000 single scheduling experiments were simulated. Average number of alternatives found for a job in CSS was 2.6. This result shows that while for relatively *small* jobs usually a few alternative executions have been found, *large* jobs usually had at most one possible execution option (remember that according to the simulation settings the difference between jobs execution time could be up to 15 times). At the same time REF algorithm at average considered more than 100 alternative executions for each job. CSS failed to find any alternative executions for at least for one job of the batch in 209 experiments; ANT - in 155 experiments. These results show that simulation settings at the same time provided quite a diverse job batch and a limited set of resources not allowing executing all the jobs during every experiment.

Figure 2 shows average job execution time (VO criterion) in a $T \rightarrow$ min, lim U optimization problem. Different limits $U_a \in \{0\%, 1\%, 4\%, 10\%, 16\%, 32\%, 100\%\}$ specify to what extent user preferences were taken into account. Two horizontal lines USER_{opt} and VO_{opt} represent practical T values when only user or VO administration criteria are optimized correspondingly.

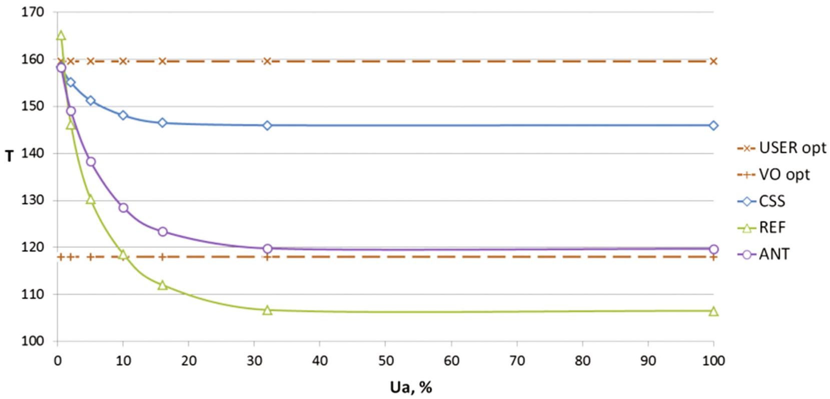


Fig. 2. Average job execution time in $T \rightarrow$ min, lim U problem

First thing that catches the eye in Fig. 2 is that REF for $U > 10\%$ provides job execution time value better (smaller) than those of $\mathrm{VO}_{\mathrm{opt}}$. However such behavior is expected as REF generates an infeasible solution and may use time-slots from more suitable (according to VO preferences) resources several times for different jobs. Otherwise ANT provided better VO criterion value than CSS for all $U > 0\%$. The relative advantage reaches 20% when $U > 20\%$ is considered. ANT algorithm graph gradually changes from $\mathrm{USER}_{\mathrm{opt}}$ value at $U = 0\%$ to almost $\mathrm{VO}_{\mathrm{opt}}$ value at $U = 100\%$ just with changing average user utility limit. Thereby ANT represents a general scheduling approach allowing balancing between VO stakeholder's criteria according to specified scenario, including VO or user criteria optimization.

A similar pattern can be observed in Fig. 3 where $C \rightarrow \max$, lim U scheduling problem is presented. However, in this case ANT advantage over CSS amounts to 10% against VO criterion.

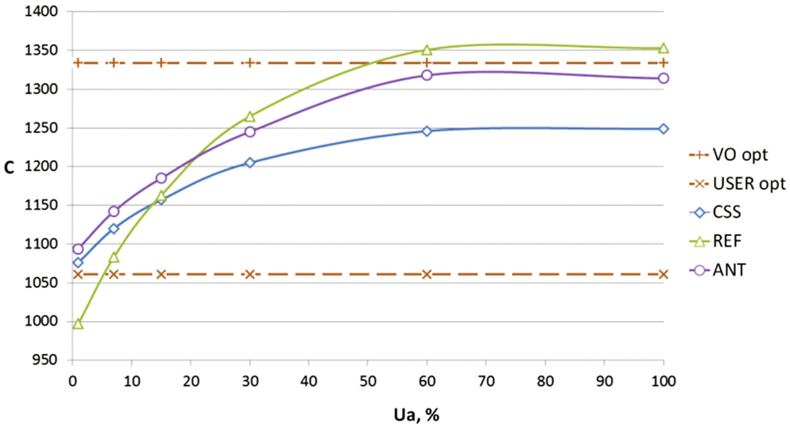


Fig. 3. Average job execution cost in $C \rightarrow \max$, lim U problem

5 Conclusions and Future Work

In this paper, we study the problem of a fair job batch scheduling with a relatively limited resources supply. The main problem arise is a scarce set of job execution alternatives which eliminates scheduling optimization efficiency. We study a heuristic scheduling scheme which generates a near-optimal but infeasible reference solution and then replicates it to allocate a feasible accessible solution. Special replication procedure is proposed which provides 2–5% error from the reference scheduling solution. The obtained results show that the new heuristic approach provides flexible and efficient solutions for different fair scheduling scenarios.

Future work will be focused on replication algorithm study and its possible application to fulfill complex user preferences expressed in a resource request.

Acknowledgments. This work was partially supported by the Council on Grants of the President of the Russian Federation for State Support of Young Scientists and Leading Scientific Schools (grants YPhD-2297.2017.9 and SS-6577.2016.9), RFBR (grants 15-07-02259 and 15-07-03401), and by the Ministry on Education and Science of the Russian Federation (project no. 2.9606.2017/BCh).

References

1. Dimitriadou, S.K., Karatza, H.D.: Job scheduling in a distributed system using backfilling with inaccurate runtime computations. In: Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 329–336 (2010)
2. Toporkov, V., Toporkova, A., Tselishchev, A., Yemelyanov, D., Potekhin, P.: Heuristic strategies for preference-based scheduling in virtual organizations of utility grids. J. Ambient Intell. Humanized Comput. **6**(6), 733–740 (2015)
3. Buyya, R., Abramson, D., Giddy, J.: Economic models for resource management and scheduling in grid computing. J. Concurrency Comput. **14**(5), 1507–1542 (2002)
4. Kurowski, K., Nabrzyski, J., Oleksiak, A., Weglarz, J.: Multicriteria aspects of grid resource management. In: Nabrzyski, J., Schopf, J.M., Weglarz, J. (eds.) Grid Resource Management. State of the Art and Future Trends, pp. 271–293. Kluwer Academic Publishers (2003)
5. Rodero, I., Villegas, D., Bobroff, N., Liu, Y., Fong, L., Sadjadi, S.M.: Enabling interoperability among grid meta-schedulers. J. Grid Comput. **11**(2), 311–336 (2013)
6. Ernemann, C., Hamscher, V., Yahyapour, R.: Economic scheduling in grid computing. In: Feitelson, D., Rudolph, L., Schwiegelshohn, U. (eds.) JSSPP, vol. 18, pp. 128–152. Springer, Heidelberg (2002)
7. Rzadca, K., Trystram, D., Wierzbicki, A.: Fair game-theoretic resource management in dedicated grids. In: IEEE International Symposium on Cluster Computing and the Grid (CCGRID 2007), Rio De Janeiro, Brazil, pp. 343–350. IEEE Computer Society (2007)
8. Vasile, M., Pop, F., Tutueanu, R., Cristea, V., Kolodziej, J.: Resource-aware hybrid scheduling algorithm in heterogeneous distributed computing. J. Future Gener. Comput. Syst. **51**, 61–71 (2015)
9. Penmatsa, S., Chronopoulos, A.T.: Cost minimization in utility computing systems. Concurrency Comput. Pract. Exp. **16**(1), 287–307 (2014). Wiley
10. Mutz, A., Wolski, R., Brevik, J.: Eliciting honest value information in a batch-queue environment. In: 8th IEEE/ACM International Conference on Grid Computing, New York, USA, pp. 291–297 (2007)
11. Blanco, H., Guirado, F., Lrida, J.L., Albornoz, V.M.: MIP model scheduling for multi-clusters. In: Euro-Par 2012, pp. 196–206. Springer, Heidelberg (2012)
12. Takefusa, A., Nakada, H., Kudoh, T., Tanaka, Y.: An advance reservation-based co-allocation algorithm for distributed computers and network bandwidth on QoS-guaranteed grids. In: Schwiegelshohn, U., Frachtenberg, E., (eds.) JSSPP 2010, vol. 6253, pp. 16–34. Springer, Heidelberg (2010)
13. Carroll, T., Grosu, D.: Divisible load scheduling: an approach using coalitional games. In: Proceedings of the Sixth International Symposium on Parallel and Distributed Computing, ISPDC 2007, p. 36 (2007)
14. Kim, K., Buyya, R.: Fair resource sharing in hierarchical virtual organizations for global grids. In: Proceedings of the 8th IEEE/ACM International Conference on Grid Computing, Austin, USA, pp. 50–57. IEEE Computer Society (2007)

15. Skowron, P., Rzadca, K.: Non-monetary fair scheduling cooperative game theory approach. In: Proceeding of SPAA 2013 Proceedings of the Twenty-Fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures, pp. 288–297. ACM, New York (2013)
16. Toporkov, V., Yemelyanov, D., Bobchenkov, A., Tselishchev A.: Scheduling in grid based on VO stakeholders preferences and criteria. Advances in Intelligent Systems and Computing, vol. 470, pp. 505–515. Springer International Publishing Switzerland (2016)
17. Toporkov, V., Toporkova, A., Tselishchev, A., Yemelyanov, D.: Slot selection algorithms in distributed computing. J. Supercomput. **69**(1), 53–60 (2014)
18. Farahabady, M.H., Lee, Y.C., Zomaya, A.Y.: Pareto-optimal cloud bursting. IEEE Trans. Parallel Distrib. Syst. **25**, 2670–2682 (2014)
19. Cafaro, M., Mirto, M., Aloisio, G.: Preference-based matchmaking of grid resources with CP-Nets. J. Grid Comput. **11**(2), 211–237 (2013)

Stability Enhancement Against Fluctuations in Complex Networks by Optimal Bandwidth Allocation

K.Y. Henry Tsang(✉) and K.Y. Michael Wong

Department of Physics, The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong
{kytsangab, phkywong}@ust.hk

Abstract. Fluctuations of resources in complex networks such as power grids are common and may cause current flows to exceed the capacity limit of the transmission links, resulting in failures. Such a failure can easily trigger cascading failures in the network. This problem can occur in power grids especially when renewable energy sources with large fluctuations such as solar and wind energy are increasingly deployed in power supply.

In this work, we introduce the discrete Green's function formulation to elucidate how resource fluctuations determine flow fluctuations in links of a network with a quadratic cost function. Furthermore, we develop an optimized bandwidth allocation scheme so as to minimize the number of failed links or the amount of excess flows in networks under fluctuations, given that the total bandwidth of the network is fixed. Compared with the conventional approach of proportionate bandwidth assignment, it is found that the optimized bandwidth allocation can highly enhance the stability of the networks against fluctuations.

1 Introduction

Modern societies rely heavily on electrical power and hence, power grids have evolved to large complex systems to satisfy the increasing power demand. Even though advanced technology was invested in enhancing the stability of power grids, widespread blackouts of different scales in power grids still occurred frequently [1]. Hence, it is important to study the cause of large blackouts and ways to enhance the stability of the networks. The occurrence of a large blackout is usually caused by the process of cascading failures [2]. Such processes usually occur when there is a link failure in the network, causing the current originally flowing in that failed link to redistribute to other links in order to satisfy the demand of the users. The redistribution of current flows generally increases the load of other links due to flow conservation. Links with increased current flows have a higher load and may break down. Those newly failed links can cause further links to break down by the same reason. Such processes can continue to spread causing large global failures. Therefore, there are increasing numbers of studies on cascading failures [3–6] and methods of controlling cascading failures in complex networks [7].

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_42

There are also new challenges about the stability in power grid systems in recent years due to the introduction of renewable energy sources. Renewable energy usually has strong fluctuations, and such increasing deployment of renewable energy in power grids can endanger the stability of the networks [8]. This is because power grids usually operate near their capacity limits and flow fluctuations can cause current flows to exceed the capacity of the transmission cables. Seeing that a small number of link failures can cause a large failure in the network, it can largely affect the stability of the network and it becomes important to study the flow fluctuations in the network [9]. One common way to increase the robustness of the network is to increase the bandwidth of links. Conventionally, the capacity layout used is the proportionate increase in initial loads [5, 6, 10]. However, it is not effective in enhancing the stability of the networks [5, 10]. As a result, there are studies on how to increase the robustness of the network by finding a better capacity layout [9, 11, 12]. From these studies, it is found that using a different capacity layout can largely increase the robustness of the networks. As fluctuations in the network can largely affect the stability of the network, in this work, we study better schemes of allocating bandwidth in the links of the network to prevent cascading failure caused by fluctuations. Furthermore, we test the bandwidth allocation schemes using a simplified power grid model in IEEE 118 bus network.

In Sect. 2, we introduce the model for the networks and discrete Green's function approach will also be described for calculating the network flow. The discrete Green's function is also used for predicting the variance of flow fluctuations given the information of resource fluctuations, verified by simulation results on some network structures in Sect. 3. In Sect. 4, we develop the optimized bandwidth allocation against fluctuations to increase the stability of the network.

2 The Model

We start with a typical resource allocation network model to study the current flows in the network. Consider a network with N nodes and each node is labeled by the index $i = 1, \cdots, N$. Each node i is given a resource Λ_i where positive Λ_i means node i has excess resource (supply node) and negative Λ_i means node i is deficient in resource (demand node). Nodes with $\Lambda_i = 0$ only relay the current flows. Similar to power grid network, we consider the resources in the network are balanced which do not have excess resources $\left(\sum_i \Lambda_i = 0\right)$. Supply nodes transport their resources to demand nodes by the current flow such that the final quantity of resources of each node becomes zero. Denote the current flow from node j to node i as $y_{ij} \equiv -y_{ji}$ and it has to satisfy the flow conservation constraint,

$$\Lambda_i + \sum\nolimits_{j \in \partial i} y_{ij} = 0, \tag{1}$$

where ∂i is denoted as the neighbors of node i. The current flows are chosen to minimize the total transportation cost function in which the cost function depends on the network model. In this work, we focus on the quadratic cost function and the total transportation cost function is given by

$$E = \sum\nolimits_{(ij)} \frac{y_{ij}^2}{2}. \tag{2}$$

The quadratic cost function is chosen to promote more uniform flows. Moreover, it represents the power dissipation in electrical networks which is mathematically equivalent to power grid network in the direct current (DC) approximation formulation. Therefore, we can use networks with quadratic cost function as simplified power grid network. To calculate the optimized current flows, we introduce the discrete Green's function formulation. The Lagrangian for optimization is given by

$$L = \sum\nolimits_{(ij)} \frac{y_{ij}^2}{2} + \sum\nolimits_i \mu_i \Big(\Lambda_i + \sum\nolimits_{j \in \partial i} y_{ij} \Big), \tag{3}$$

where μ_i is the Lagrangian multiplier and it can be interpreted as the chemical potential of node i. Optimizing L with respect to y_{ij}, one can obtain the current flows as

$$y_{ij} = \mu_j - \mu_i. \tag{4}$$

The current flow y_{ij} on a link can be interpreted as it is driven by the chemical potential difference between node i and j. Therefore, the problem now becomes finding the chemical potential of each node. Substituting Eq. (4) into the flow conservation constraint, the chemical potential μ_i can be expressed in terms of the μ_j of its neighbors,

$$\mu_i = \frac{1}{d_i} \Big(\Lambda_i + \sum\nolimits_{j \in \partial i} \mu_j \Big), \tag{5}$$

where d_i is the degree of node i. Instead of using iteration in Eq. (5), one can obtain the chemical potential by the formulation of discrete Green's function. Introduce the Laplacian matrix L from graph theory,

$$L = D - A, \tag{6}$$

where D is a $N \times N$ diagonal matrix with diagonal elements D_{ii} equal to d_i and A is the adjacency matrix. Rewrite Eq. (5) using L, one can obtain the following (in matrix notation) as

$$-\sum\nolimits_{j \in \partial i} \big(\mu_j - \mu_i \big) = \sum\nolimits_j L_{ij} \mu_j = \Lambda_i \Leftrightarrow L\mu = \Lambda, \tag{7}$$

where μ and Λ are column matrix with i^{th} element as the chemical potential and resources of node i, respectively. From Eq. (7), we can obtain the chemical potential of the nodes by solving the Laplace equation. However, L is singular and we need to use the generalized inverse of L to solve Eq. (7). The generalized inverse of the Laplacian matrix is called the discrete Green's function, G [13]. As a result, chemical potentials and the current flows are given by

$$\mu = G\Lambda \Rightarrow y_{ij} = \sum\nolimits_i \big(G_{jl} - G_{il} \big) \Lambda_l. \tag{8}$$

3 Flow Fluctuations Induced by Resource Fluctuations

To calculate the flow fluctuations induced by the resource fluctuations, we start by introducing resource fluctuations in the network and compute the corresponding changes in current flows. Suppose Λ_i^0 is the resources of node i without fluctuation and the resource fluctuations is $\delta\Lambda_i$, the resource becomes

$$\Lambda_i = \Lambda_i^0 + \delta\Lambda_i. \tag{9}$$

Since the current flows are determined by the resource of the nodes, the current y_{ij} will change according to

$$y_{ij} = y_{ij}^0 + \delta y_{ij}, \tag{10}$$

where y_{ij}^0 is the initial current without fluctuation and δy_{ij} is the flow fluctuations which we are interested in. From the discrete Green's function formulation, we can see that the change in current flows are given by

$$\delta y_{ij} = \sum\nolimits_l \left(G_{jl} - G_{il}\right)\delta\Lambda_l. \tag{11}$$

The mean $\left\langle\delta y_{ij}\right\rangle$ is obtained by taking average of Eq. (11) and the variance $\left\langle\delta y_{ij}^2\right\rangle$ is obtained by taking the squares and then averaging on both sides of Eq. (11),

$$\left\langle\delta y_{ij}^2\right\rangle = \sum\nolimits_l \left(G_{jl} - G_{il}\right)^2\left\langle\delta\Lambda_l^2\right\rangle, \tag{12}$$

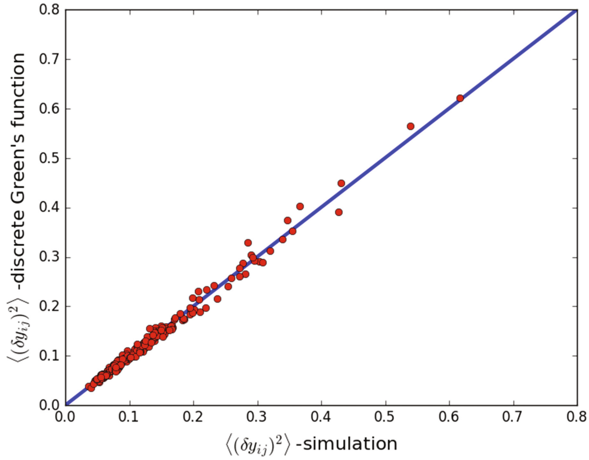


Fig. 1. The y-axis is obtained by Eq. (12) while the x-axis is the simulation result. The simulation is done in ER network with average degree ≈5 and $N = 100$. It shows that the simulation result gives a good agreement with the estimation using discrete Green's function

where we have assumed the fluctuations in resources are uncorrelated with each other (i.e. $\left\langle \delta\Lambda_i \delta\Lambda_j \right\rangle = \delta_{ij} \left\langle \delta\Lambda_i^2 \right\rangle$ where δ_{ij} is the Kronecker delta function) for simplicity.

After obtaining the discrete Green's function approach to estimate the flow fluctuations, we test the accuracy of the estimation by simulations in ER network (with 100 nodes and average degree ≈5). In the simulation, the resource fluctuations are uncorrelated and follow the Gaussian distribution with mean $\langle \delta\Lambda_i \rangle = 0$ and variance $\left\langle \delta\Lambda_i^2 \right\rangle = 1$. The flow fluctuations are then follows a Gaussian distribution with mean $\left\langle \delta y_{ij} \right\rangle = 0$ as $\langle \delta\Lambda_i \rangle = 0$. Figure 1 shows the simulation result and it shows that using discrete Green's function can give a good estimation of the variance of the flow fluctuations.

4 Optimal Bandwidth Allocation Against Fluctuation

4.1 Proportional Bandwidth Allocation

The bandwidth of a link is a terminology from communication networks which we generalize it for convenience in a network and it refers to the maximum amount of current that can flow in that link before it becomes overloaded. Since the bandwidth of a link determines the amount of current flow that can pass through it, it usually has the capacity withstanding the current flow of the network in the steady state with some tolerance. The reason for the tolerance is to prevent the overload from a sudden increase or fluctuations in current flows. For simplicity, the bandwidth in the network is written as

$$L_{ij} = L_{ij}^0 + \Delta L_{ij}, \tag{13}$$

where $L_{ij}^0 = \left| y_{ij}^0 \right|$ and $\Delta L_{ij} \geq 0$ represents the tolerance. The traditional proportional bandwidth has the form of

$$L_{ij} = (1 + \epsilon) L_{ij}^0, \tag{14}$$

where ϵ is the tolerance factor. In practice, increasing the capacity of bandwidths will increase the cost of constructing and is usually desired to be small. Thus, we are interested in the total construction cost $C = \sum_{(\mathrm{ij})} L_{ij}$. However, for a given total cost C, allocating bandwidth using the form of Eq. (14) is not effective against flow fluctuations. To illustrate this, simulations in random networks (with 100 nodes and average degree ≈5) using the proportionate bandwidth allocation are used for demonstration. When there are fluctuations in the current flows, the current may exceed the bandwidth capacity and the link considered overloaded when $\left| y_{ij} \right| > L_{ij}$. The fraction of failed links is used as a measurement of the effectiveness of bandwidth allocation scheme against fluctuations. In the simulation, the resource fluctuation $\delta\Lambda_i$ follows a Gaussian distribution independently with mean equal 0 and variance equal $0.01 \left| \Lambda_i^0 \right|^2$ (Λ_i^0 follows a Gaussian distribution with mean equal 0 and variance equal 100). The simulation results are obtained from 100 samples averaging. From Fig. 2, it can be seen that the

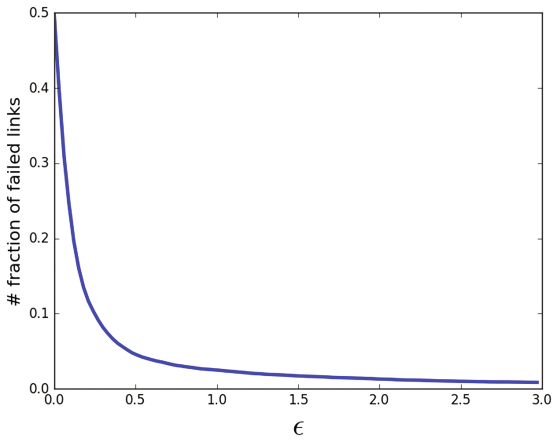


Fig. 2. Simulation results of the proportionate bandwidth allocation scheme in ER network with average degree ≈5 and $N = 100$. The vertical axis represents the fraction of failed links while the horizontal axis represents the tolerance factor ϵ. Note the presence of long tails indicating that there are still some failed links in the network even for a large tolerance factor

proportionate bandwidth allocation scheme is not effective against fluctuations in the sense that there are still some failed links even when ϵ is large. The main reason for the inefficiency of proportionate bandwidth allocation against fluctuations is the existence of links with small y_{ij}^0 and the tolerance factor increases independently with flow fluctuations. In the proportionate bandwidth allocation scheme, if y_{ij}^0 is small, the resulting bandwidth change $\left(\epsilon L_{ij}^0\right)$ is small even for large value of ϵ, whereas the flow fluctuations are often much larger.

4.2 Proposed Optimized Bandwidth Allocation

Seeing that the proportionate bandwidth allocation is not robust against flow fluctuations, it is essential to develop a better allocation of bandwidths to enhance the stability of the network against fluctuations. Therefore, we are interested in finding ΔL_{ij} from a given total cost which is a constraint given by

$$\sum\nolimits_{(ij)} \Delta L_{ij} = C. \tag{15}$$

The allocation of bandwidth resources can be formulated as an optimization problem which we need to find the additional bandwidths ΔL_{ij} that satisfy some specific objective function under the constraint Eq. (15). We propose two objective functions for optimization that can measure the robustness of the network - the total number of failed links and the total amount of excess current in the network.

For simplicity, we assume that the resource fluctuations are uncorrelated with each other and follow Gaussian distributions with mean equal 0. The probability of a link to fail is determined by the probability distribution of the flow fluctuations (which is a Gaussian distribution). Considering δy_{ij} in the direction of increasing y_{ij}^0, it is sufficient to consider the fraction of distribution with $\delta y_{ij} > \Delta L_{ij}$. Thus, the total number of failed links is given by

$$F = \sum\nolimits_{(ij)} \int_{\Delta L_{ij}}^{\infty} P_{ij}(y)dy, \tag{16}$$

where $P_{ij}(y)$ is the distribution of the fluctuating flow in link (ij). The construction costs for the minimization are

$$\sum\nolimits_{(ij)} \Delta L_{ij} = C \text{ and } \Delta L_{ij} \geq 0. \tag{17}$$

The Lagrangian for the optimization problem is given by

$$L = \sum\nolimits_{(ij)} \int_{\Delta L_{ij}}^{\infty} P_{ij}(y)dy - \lambda\left(\sum\nolimits_{(ij)} \Delta L_{ij} - C\right) + \sum\nolimits_{(ij)} \gamma_{ij} \Delta L_{ij}, \tag{18}$$

where λ is the Lagrange multiplier and γ_{ij} is the Kuhn-Tucker multiplier. For Gaussian distribution $P_{ij}(y), \Delta L_{ij}$ that can minimize the total number of failed links is given by

$$\Delta \mathrm{L}_{ij} = \begin{cases} 0, & \mu \leq \sqrt{2\pi\left\langle \delta \mathrm{y}_{\mathrm{ij}}^2 \right\rangle} \\ \sqrt{\delta \mathrm{y}_{\mathrm{ij}}^2}\left[2\ln\left(\mu\Big/\sqrt{2\pi\left\langle \delta \mathrm{y}_{\mathrm{ij}}^2\right\rangle}\right)\right]^{1/2}, & \text{otherwise} \end{cases} \tag{19}$$

where $\mu \equiv 1/\lambda$ is determined by Eq. (15). The conditional form of Eq. (19) is due to Kuhn-Tucker condition.

Another objective function we propose is the total amount of excess current flows. Excess current flows are defined as the amount of current flow exceeding the bandwidth and thus, the function measuring the total excess current flow in the network is given by

$$F = \sum\nolimits_{(ij)} \int_{\Delta L_{ij}}^{\infty} \left(y - \Delta L_{ij}\right) P_{ij}(y)dy. \tag{20}$$

The optimal bandwidth allocation scheme that can minimize the total excess current is obtained by minimizing Eq. (20) subject to the total cost constraint as Eq. (17). After optimization, one can obtain

$$\Delta L_{ij} = \epsilon' \sqrt{\left\langle \delta y_{ij}^2 \right\rangle} \tag{21}$$

where $\epsilon' = \sqrt{2}\mathrm{erfc}^{-1}(2\lambda)$ and it can be calculated by substituting ΔL_{ij} into Eq. (18).

4.3 Simulation Results and Discussion

After having the two bandwidth allocation schemes, we compare their performances with simulations. Since the network model we use can be related to the power grid in the DC approximation, we test the bandwidth allocation in the simulation related to real life power grid networks, such as the IEEE 118-bus test case [14]. The IEEE 118-bus network is a usual power grid model testing network for simulation. For simplicity, networks with bandwidths allocated by the proportionate scheme are denoted as *proport*, and those obtained by minimizing respectively the number of failed links and the excess flows are denoted as *minlink* and *minflow*. These bandwidth allocation schemes are compared using the same total construction cost. The total investment cost is parameterized by the tolerance factor $\epsilon = C / \sum_{(ij)} L^0_{ij}$. In the simulation, the resource fluctuations $\delta\Lambda_i$ of node i follows Gaussian distributions with means equal 0 and variance equal $0.01 \left| \Lambda^0_i \right|^2$. The variance of flow fluctuations calculated by Eq. (12) is used to assign the bandwidth according Eqs. (19) and (21). The simulation results are obtained by 100 independent samples average. Figure 3 shows the simulation results of the fraction of failed links in the network or the total amount of excess current as a function of the tolerance factor. As expected, the simulation results show that using the bandwidth *minlink* has the least fraction of failed links while *minflow* has the least amount of excess current under flow fluctuations. Remarkably, since *minlink* and *minflow* can approach to zero in fraction of failed links (also in the total excess current) using a relatively small amount of investment cost as compared with the proportionate bandwidth allocation scheme, they can give much better stability to the networks under fluctuations.

One of the main reasons for the effectiveness of the two new bandwidth allocation schemes against fluctuations is that the increase in bandwidth has considered the magnitude of fluctuations. For example, *minlink* allocate more resources to those links with small flow fluctuations (i.e. $\left(\left\langle \delta y^2_{ij} \right\rangle \right)^{1/2} < \mu/2\pi$) when the given total cost is small (unlike the proportionate bandwidth allocation scheme which allocates the bandwidth resources to every link independent of the magnitude of flow fluctuations). When C increases, the non-linearity of Eq. (19) tends to distribute more bandwidth resources to links with moderate flow fluctuations to save the majority of links. With such a distribution of bandwidth resources, it can minimize the total number of failed links in the network with fluctuations by a given investment cost. Furthermore, *minflow* increases the bandwidths in the links proportional to the magnitude of the standard deviation of the flow fluctuations taking into consideration the effect of fluctuations.

Note that from Fig. 3, there exist some points with discontinuous slope in the curve of *minlink*. The discontinuities come from the jumps in ΔL_{ij} as derived in Eq. (19) and it can be illustrated by the following example. Suppose a link k in the network has a flow fluctuation $\langle \delta y^2_k \rangle$ which is much larger than other links' flow fluctuations, then according to Eq. (19), link k does not obtain any bandwidth resources. Consider the scenario of gradually increasing the total bandwidth with μ increasing accordingly. μ is large but still smaller than $\sqrt{2\pi \langle y^2_k \rangle}$, all the links except link k receive much bandwidth resources and most of the links do not have excess current. However, the large excess

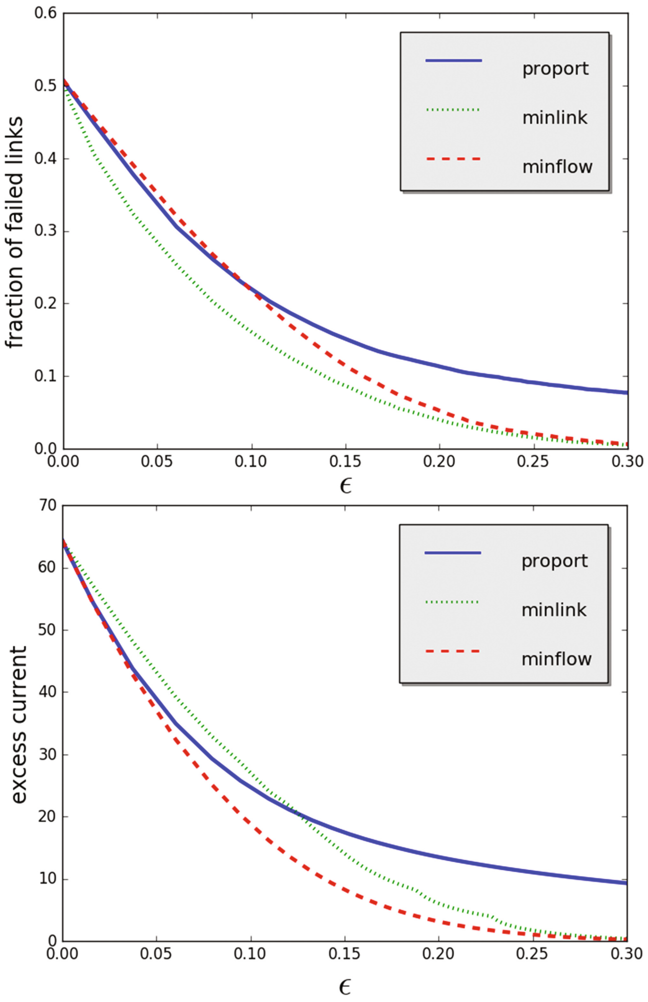


Fig. 3. Simulation results on IEEE 118-bus network (with 100 sample average). The x-axis is the tolerance factor. The figures show that *minlink* has the least fraction of failed links while *minflow* has the least amount of excess current. Furthermore, comparing with *proport*, both *minlink* and *minflow* gives higher network stability under fluctuations with the same total cost

current in link k remains the same and the decreasing rate of total excess current is slow. When μ continues to increase and start to be larger than $\sqrt{2\pi\langle y_k^2\rangle}$, the link k starts to receive bandwidth resources and therefore, the excess current in link k decrease suddenly. This results in a discontinuity of slope in the curve *minlink* in Fig. 3.

5 Summary

In summary, we have formulated the discrete Green's function for finding the current flows in the transportation network with a quadratic cost function which can be treated as simplified power grid networks in the DC approximation. We also estimate the current flow fluctuations induced by resource fluctuations using the discrete Green's function which gives a high accuracy in the simulations result. Furthermore, we have shown that using the conventional proportionate bandwidth allocation cannot effectively increase the robustness of the networks against fluctuations. To properly allocate the bandwidths, we developed the optimal bandwidth allocation schemes that can minimize the total number of failed links or total excess current in the networks under fluctuations. The simulation results show that the optimized methods of allocating bandwidth can effectively enhance the stability of the networks against fluctuations compared with the proportional bandwidth allocation.

Acknowledgement. We thank David Saad for fruitful discussions. This work is supported by the Research Grants Council of Hong Kong (grant numbers 605183 and 16322616).

References

1. http://www.nerc.com/dawg/database.html. Information on electric systems disturbances in North America
2. Lai, Y.-C., Motter, A.E., Nishikawa.: Attacks and cascades in complex networks. In: Complex Networks, pp. 299–310. Springer, Heidelberg (2004)
3. Dobson, I., Carreras, B.A., Lynch, V.E., Newman, D.E.: Complex systems analysis of series of blackouts: cascading failure, critical points, and self organization. Chaos Interdisc. J. Nonlinear Sci. **17**(2), 026103 (2007)
4. Crucitti, P., Latora, V., Marchiori, M., Rapisarda, A.: Efficiency of scale-free networks: error and attack tolerance. Physica A Stat. Mech. Appl. **320**, 622–642 (2003)
5. Motter, A.E., Lai, Y.-C.: Cascade-based attacks on complex networks. Phys. Rev. E **66**(6), 065102 (2002)
6. Crucitti, P., Latora, V., Marchiori, M.: Model for cascading failures in complex networks. Phys. Rev. E **69**(4), 045104 (2004)
7. Bienstock, D., Chertkov, M., Harnett, S.: Chance-constrained optimal power flow: riskaware network control under uncertainty. SIAM Rev. **56**(3), 461–495 (2014)
8. Harrison, E., Saad, D., Wong, K.Y.M.: Message passing for distributed optimisation of power allocation with renewable resources. In: 2016 2nd International Conference on Intelligent Green Building and Smart Grid (IGBSG). IEEE (2016)
9. Heide, D., Schäfer, M., Greiner, M.: Robustness of networks against fluctuation-induced cascading failures. Phys. Rev. E **77**(5), 056103 (2008)
10. Albert, R., Jeong, H., Barabási, A.-L.: Attack and error tolerance of complex networks. Nature **406**(6794), 378–382 (2000)
11. Schäfer, M., Scholz, J., Greiner, M.: Proactive robustness control of heterogeneously loaded networks. Phys. Rev. Lett. **96**(10), 108701 (2006)

12. Li, P., Wang, B.-H., Sun, H., Gao, P., Zhou, T.: A limited resource model of fault-tolerant capability against cascading failure of complex network. Eur. Phys. J. B **62**(1), 101–104 (2008)
13. Chung, F., Yau, S.-T.: Discrete Green's functions. J. Comb. Theory Ser. A **91**(1), 191–214 (2000). ISSN 0097-3165, http://dx.doi.org/10.1006/jcta.2000.3094
14. http://www.ee.washington.edu/research/pstca/. The IEEE 118 bus-network model is a standard test system

The Scope of the Collected Data for a Holistic Risk Assessment Performance in the Road Freight Transport Companies

Agnieszka Tubis and Sylwia Werbińska-Wojciechowska(✉)

Wroclaw University of Technology, 27 Wybrzeze Wyspianskiego Street,
Wroclaw, Poland
{agnieszka.tubis,sylwia.werbinska}@pwr.edu.pl

Abstract. In the presented paper, authors focus on the issues connected with passenger transportation companies risk assessment. Following this, in the article the authors investigate the problem of information needs that make possible a full risk analysis performance for freight transport companies. Thus, the holistic approach in risk assessment for road transport companies is discussed. Later, there is presented a short literature review connected with information systems dedicated for risk management performance. This gives the possibility to define the main data reporting system that meets the requirements of risk assessment performance. Article ends with some conclusions and directions for future research.

Keywords: Transportation system · Risk assessment · Holistic approach · Data collection

1 Introduction

The transportations systems are a very complex organizations composed of a wide array of infrastructures such as terminal facilities, travel ways, transportation fleets, and information systems. Such systems are decentralized and open, thus provide easy and reliable access for many users. As a result, transportation systems are exposed to many risks of external and internal nature [24, 25, 38].

Due to the negative consequences of such risks occurrence, it is crucial to recognize the sources of risks, helping to maintain continuity and timeliness of the transport process performance. Thus, transport companies should implement a risk management system and regularly carry out risk analysis, which is based on identification of potential hazards or situations or conditions that lead to threats. These risks are associated with the occurrence of events, both random and non-random ones [26, 42].

One of the most important problems in the area of risk management of any company is to acquire, maintain and aggregate data across diverse trading units. Thus, the design of an information system depends on a risk measurement methodology that a firm chooses [18]. Following this, there is a trade-off between the accuracy of the resulting measures of risk and the burden of computing them with the use of accurate IT technology [10].

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_43

Following this, in the article authors focus on the issues connected with risk management in road transport processes performance. The aim of the article is to analyse the problem of information needs that make possible a full risk analysis performance for freight transport companies. As a result, in the next Section, authors focus on the presentation of the issues on holistic approach in risk assessment for road transport companies. Then, there is provided a brief overview of the literature in the area of information systems and information needs for risk management performance. This gives the possibility to investigate the data reporting system being used in the chosen road freight transport company. The information system is analysed taking into account its usability in the area of full risk assessment process performance. The article concludes with a summary and guidelines, including directions for further research.

2 Holistic Approach in Risk Assessment for Road Transport Companies

Currently, there is no one, unified and commonly used definition of risk term [6]. We can even state that the underlying concepts of risk are hard to define and even harder to assess [22]. In recent decades, we have observed this term being applied to many research areas, like decision theory, management, emergency planning, or critical structures operation, including transport systems performance [41]. The historical development trends of risk concept are discussed e.g. in [3, 6]. The risk perspectives review and discussion are given e.g. in [5, 7–9].

One of the most often cited risk term definition is given in PN-ISO 31000 standard [31], where risk is defined as *effect of uncertainty on objectives*. A brief summary of classification of risk definitions is given e.g. in [6, 19]. Based on this, the same standard defines risk management as *coordinated activities to direct and control an organization with regard to risk*. The developed definition is very general. Thus, in order to effectively manage any organization, the new concept is introduced and promoted - Enterprise Risk Management. One of the most popular definitions of Enterprise Risk Management concepts (ERM) used in the literature is the one provided by COSO II standard. According to COSO II standard [13] Enterprise Risk Management is defined as *a process, effected by an entity's board of directors, management and other personnel, applied in strategy setting and across the enterprise, designed to identify potential events that may affect the entity, and manage risk to be within its risk appetite, to provide reasonable assurance regarding the achievement of entity objectives*. According to COSO II, an organization's ERM system should be geared toward achieving the following four objectives: (1) Strategy: high-level goals, aligned with and supporting the organization's mission. (2) Operations: effective and efficient use of the organization's resources. (3) Reporting: reliability of the organization's reporting system. (4) Compliance: organizational compliance with applicable laws and regulations.

The proper implementation of ERM conception also influences the risk assessment processes being performed in the chosen company. The risk assessment is an essential and systematic process that is a part of risk management which aims at identifying, assessing the risks and planning the actions to deal with the risks [36]. However, there is a diversity in risk analysis procedures and techniques that may be used in this area (for review we recommend reading e.g. [4, 16, 21, 28, 29]).

Referring to the previously presented definitions and taking into account the process perspective, the starting point for risk assessment performance should be to identify threats that may be the cause of failure to achieve the objectives in the passenger transportation services performance. A holistic approach assumes that the area of analysis will cover different levels of performed process, like e.g. technical elements, human resource, as well as legal and organizational issues. Risk assessment should therefore be preceded by a process analysis that allows identifying potential adverse events. The risk assessment is usually performed only at certain time points. The main challenge in this field is assessing all the risks in a system or organization what is determined by the proper information system support. Having timely information is a key issue to an effective ERM program and risk assessment performance.

Research conducted by the authors clearly show that for the transport company management processes, the current risk assessment models used in the area of transport processes performance are insufficient [42]. During operational business performance, the managers are exposed to the presence of various risks, which are different than those described in the scientific research. For the purposes of decision-making processes, it is necessary to build a model of a risk assessment taking into account the process approach, consistent with the concept of Enterprise Risk Management. This problem is also underlined in the current EU research projects focused on road transport networks security issues. The short overview of the current EU-funded research into transport security is presented e.g. in [12]. One of the interesting research projects is the SERON project [43]. This project is focused on the investigation of the impact of possible manmade attacks on the transport network (see e.g. [23, 43] for more information).

Proposed process approach, in accordance with ISO 31000 standards, implies a holistic approach to risk assessment in the company. This means that the identification of potential hazards is done by the way of a process analysis, which includes the analysis of used resources (elements at the input to the process), the course of the process and the expected final result. Process approach also assumes that the process is carried out in a certain environment, which affects its performance. For this reason, the sources of potential hazards are identified as both internal and external ones. Thus, the risk assessment takes into account financial, technical, informational, social and organizational issues.

In the case of road transport companies, the performed risk analysis is focused on the identification of the maximum number of possible adverse events, which may accompany the two defined above performed processes. The defined procedure involves the evaluation the main steps shown in Fig. 1.

In the case of a transport company providing services at international transport level, the risk assessment should be carried out for each direction of the movements separately. This is due to the fact that each export or import freight transport to defined countries is connected with the occurrence of general hazard events and specific risks associated with a particular direction. Not taking into account the specific nature of transport process performed on the given direction and reducing the risk assessment to the general level for all the movements, significantly reduce the complexity of the

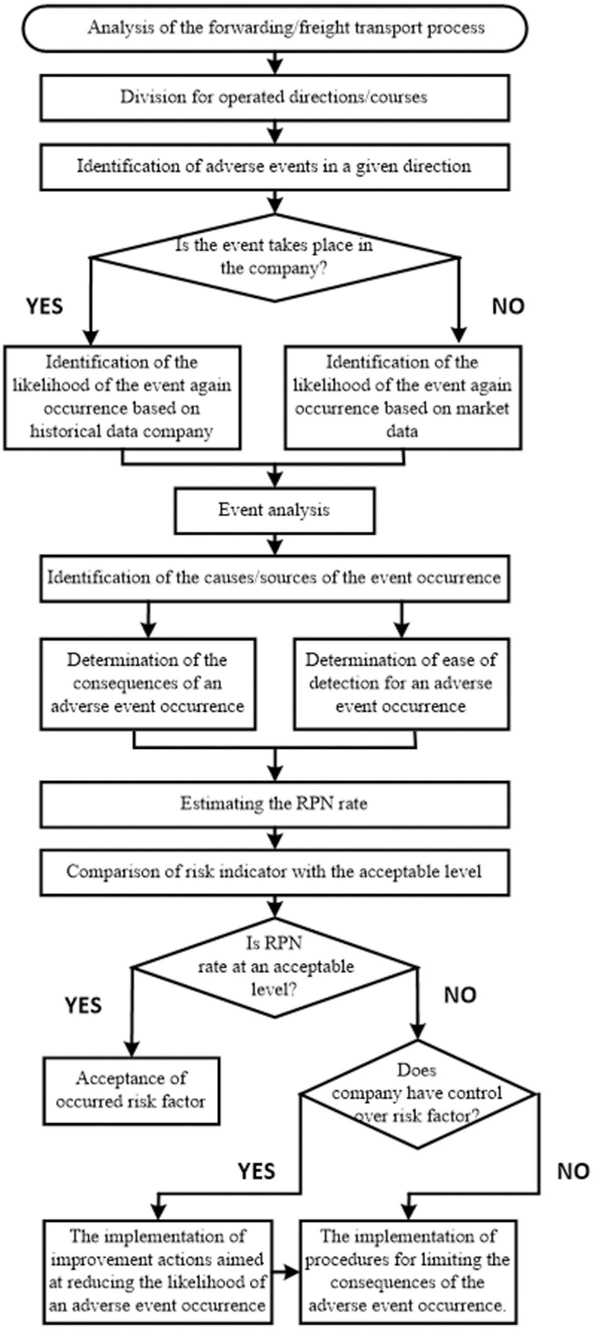

Analysis of the forwarding/freight transport process
Division for operated directions/courses
Identification of adverse events in a given direction
Is the event takes place in the company?
YES
NO
Identification of the likelihood of the event again occurrence based on historical data company
Identification of the likelihood of the event again occurrence based on market data
Event analysis
Identification of the causes/sources of the event occurrence
Determination of the consequences of an adverse event occurrence
Determination of ease of detection for an adverse event occurrence
Estimating the RPN rate
Comparison of risk indicator with the acceptable level
Is RPN rate at an acceptable level?
YES
NO
Acceptance of occurred risk factor
Does company have control over risk factor?
YES
NO
The implementation of improvement actions aimed at reducing the likelihood of an adverse event occurrence
The implementation of procedures for limiting the consequences of the adverse event occurrence.

Fig. 1. Risk assessment procedure for road freight transport companies

analysis and reduce its effectiveness. This will decrease its usability in managers decision-making processes by providing information needs are not fully met. The more detailed analyses, however, generates the need for more comprehensive knowledge base accumulated in the company and sets specific requirements for the reporting system. Following this, in the next Section, the comprehensive literature review in the area of information systems developed for risk management performance is given. This gives the possibility to define the basics for data reporting system for the needs of risk assessment in the selected road freight transport company.

3 Information Systems for Risk Management – Literature Review

The issue of the role of information systems in risk management is widely discussed in the current literature (see e.g. [17] for the review).

The problem of designing an information system for risk management is connected with providing managers with the data they need to make a proper business decisions. Currently, most managers want four things from their risk management information systems [17]: (a) calculate value at risk; (b) perform scenario analyses; (c) measure current and future exposure to each counterparty; (d) give the possibility to aggregate information across various groups of risks, product types, and across subsets of counterparties. The detailed risk management guide that investigates the main requirements in the area of properly defined information systems is given in [37].

Currently, widely analysed in the literature is a complex solution for effective risk management performance that regards to RIMS implementation. According to [15], RIMS is as *a technology system that enables to capture, manage and analyse all organization's risk and insurance data in a single, secure system*. This solution is generally introduced in an insurance sector (see e.g. [18, 34]). Other research area regards to security risk management being focused also on the reliability and safety of stored data. The issues of information systems security risk management are overviewed in work [1], and the effectiveness of safety management systems is given in e.g. [40].

Moreover, the evolution of information technologies that will provide organizations with sufficient and reliable data in these fields is discussed in [30].

The current knowledge on the issues of risk and safety in the transport sector is reviewed in work [33]. Authors in their publication mostly focused on the safety issues in the transport sector and provide the introduction to the RISIT (risk and safety in the transport sector) research programme performed in Norway. However, the conclusions are more general and regard to any transportation system. Authors underline that little research has been done in this area, and a number of research subjects should be considered, including the information systems effective designing and implementation.

Information issues of risk and safety management in the transport sector mostly regard to the supply chain management (see e.g. [11, 20]), public transport performance (see e.g. [14]), or information safety (see e.g. [14, 35]). In [11] authors focus on LNG transportation systems vulnerability and resilience analyses performance. They focus on marine LNG transportation system taking into account the possibility of quantitative data about the cost of disruptions and the effects of mitigating measures. The problem is also analysed in [2], where authors also focus on marine transportation industry and discuss the possibility of ERM conception implementation. Later, in work [20], authors investigate the scope of information systems dedicated for transport logistics.

The public transit in the view of cybersecurity considerations is the authors' topic of interest in recommended practice report [14]. In this work authors underline that control and management systems are dependent on information technology what causes their vulnerability to increasingly sophisticated direct and indirect cyberattacks. Following this, in the next work [35] authors describes a plan for improving industrial control systems cybersecurity across all transportation modes: aviation, highway, maritime, pipeline, and surface transportation. The main assumptions given in this plan are also satisfied e.g. in works [27, 32, 39, 44, 45].

Taking one step further, in the risk management concept the main role takes an undesired event occurrence. More detailed analysis requires also description of type of hazard, cause, consequence and way of removing the problem. In this approach, due to freight transportation system one may recognize:

- hazard – possibility of developing an undesired event,
- undesired event - not completing transportation task or delay greater than acceptable by a customer, any charm influencing freight/driver/environment in a negative way (e.g. health injury),
- risk – possibility of running a hazard expressed by frequency of undesired event over given period or travelled distance multiplied by amount of losses.

Measures of undesired event relate to:

- measure of possibility- frequency over given time of developing an undesired event,
- measure of losses- mean number of fatalities regarding to undesired event, degree of disability, time of delay, monetary equivalent for lost time or delay or equivalence of not gained profit because of not completed transportation task.

In the approach described in the paper, the following variables are expected to collect:

- date and time of an event, mileage, number of cycles, total transported goods since the previous event, etc.,
- place of an event, terrain topography,
- elements of transportation system and infrastructure taking part in the event, relations to other transportation systems,
- number of casualties other losses (fatality, injuries, number of people delayed),
- duration of disturbances for traffic: direct (closing a road), indirect (detour),
- event consequences, loss of properties, loss of transportation mean, loss of technical infrastructure and environment,
- event cause,
- way of repair, clear away consequences.

Reliability and safety data concerning regional or national transportation system may be obtained at various levels of management. The highest level of administration provides general assessment of safety in form of reports or statements usually issued yearly.

According to road transportation there are several data sources, though data are processed and concerns statistical image of the process. The main are presented below.

Reports issued by Polish Police are ordered in months and years, concerns state regions, road users, distribution of accidents due to time, terrain, cause, consequence and severity (victims).

The national program concerning life protection of road users GAMBIT exists since 2001. Now it was introduced GAMBIT 2020 with main target as decreasing in half number of victims until 2020. Program GAMBIT is operated by State Board of Road Safety. This institution also announces annual and half-yearly reports covering cross sectional analysis of accidents and actions undertaken on road safety improvement.

Web service of GDDKiA (General Headquarters of Polish Roads and Highways) provides information about local traffic restrictions and disturbances, like road building and rebuilding or road and infrastructure failures.

State Fire Service collects data concerning all rescue actions involving fire brigades. Data are collected in form of very precise data base. Application of these data to reliability and safety assessment is possible after processing and being restricted to events developed in transportation system.

However, based on the given databases it is very difficult to achieve description of single accident to perform investigation directed on cause-consequence analysis that is valuable in reliability and risk/safety approach. In many cases data are fragmentary and do not provide important variables. Value of data is continuously improved but unfortunately very often still not credible due to reorganization and bad quality of informatics' system regarding data exchange.

The presented databases are the external sources of information. In the next Section, authors focus on the data reporting system that stores the main internal data being necessary for risk assessment performance in the selected road freight transport company.

4 Data Reporting System for the Needs of Risk Assessment in the Selected Road Freight Transport Company

The main processes analysed in a transport company can vary greatly depending on, among others, the type of carried cargo, the entity performing the carriage, the required security of cargo, route. The data reporting system should take into account all the parameters of the process that requiring registration for future risk assessment. The scope of this article does not allow carrying out a detailed analysis of all cases to be assessed. For this reason, the presented results are limited only to the analysis at a general level, concerning risk factors of a universal nature that may arise in the implementation of each type of transport.

Data reporting system should be implemented to enforce data entry on a regular basis before, during and after the completion of the process. However, this caused the pressure on employees to shorten the time for data registration. Thanks to the IT tools used in a company, the reporting process should be automated as much as possible. However, this requires the prior definition of parameters that describe the processes, determination of the rules of their order, and above all, the collection of data in an electronic database (currently some of the information are the know-how of individual employees).

It is extremely important to organize data collection process in order to improve the further analysis performance and the proper distribution of results. Suitable systematization of the administered data should be performed in accordance with the accepted classification rules. For transport companies, the authors suggest the following groups of defined data for risk assessment performance: data (a) relating to drivers, (b) relating to freight forwarders, (c) relating the traders, (d) relating to subcontractors of transport services, (e) relating to customers, (f) regarding the vehicles, (g) relating to the implementation of the process, (h) relating to finance, including costs and revenues. The use of such classification allows for easy and non-confrontational assignation of responsibility for the collection of individual data by company's departments (Table 1).

Table 1. The main group of gathered data with assigned departments being responsible for their collection

Gathered data	Responsible department
Relating to drivers	Director of Transport department/HR department
Relating to freight forwarders	Shipping director/HR department
Relating to traders	Director of Sales department/HR department
Relating to subcontractors of transport services	Shipping department
Relating to customers	Sales department
Relating to vehicles	Transport department
Relating to processes implementation	Shipping department/Transport department
Relating to finance	Accounting department

The analysed company provides transport-forwarding services, mainly in road transport. It operates the cargo loads transported both in the domestic and international distributions. Currently, the company has 70 own vehicles, which are used primarily to serve regular customers. At the same time, it cooperates regularly with selected smaller carriers. The carriers are outsourced to other regular services (those that cannot be operated with company's own fleet) and all additional appearing orders, usually of single nature. In the situation of inability to perform emerging additional orders by regular collaborators, such transports are subcontracted to other carriers, usually acquired through the freight exchange. The company belongs to the SME sector, but in the last three years one can observe its intense development. As a result, the carrier has enjoyed steady growth in participation in operated markets, accompanied by expansion of the organization.

With the development of the company, the Management Board saw the need to implement the risk management system. The first step in the implementation of this concept has become a verification of the existing reporting system and its evaluation in terms of the complexity of the supplied data, required for the risk assessment process performance. Currently, the company uses a dedicated class software TMS

(Transportation Management System), but the Management Board is aware that not all information is recorded in this system. The reason is connected with the lack of appropriate procedures and limited measurement system, currently used to assess the effectiveness of the performed processes. Moreover, the current employee incentive system does not take into account the quality of the reported data.

On the basis of the conducted observations and accompanying interviews with senior managers in the audited company, there was defined the need for information to support the risk assessment process for freight forwarding activities and transport. Required data are grouped according to the classification rules and are shown in Table 2. At the same time, these data are ranked under heading of:

- the degree of control (**control level**) of the unwanted event occurrence described by the value (1 - lack of control; 2 - partial control; 3 - full control) - conducted grouping will impose further path for the risk assessment procedure performance,
- nature of the data (**nature**), defining their use in the process of analysis (quantitative data (I) and non-quantitative (NI)). Quantitative data will serve to estimate the possibility of the adverse event occurrence likelihood, but also the estimation of its consequences. They are used for quantitative analyses and measurements associated with risk assessment performance. Non-quantitative data are primarily used to define the causes and consequences of the event occurrence.

Then, on the basis of the performed preliminary analyses, there is defined the current level of reporting system in the field of required data supply. The evaluation concerned the scope of collected data and the form of their registration. The data collection was assessed in a 3-point scale (**the scope**): 1 - not available, 2 - limited/insufficient number of data, 3 - range adequate to the needs. **Registration forms** are structured as follows: (a) complete knowledge database (BWS) – whole recorded currently in the database, regularly updated, standardized and reliable data; (b) relative knowledge database (WWS) - data entered into the system, but no indication of the person responsible for taking care of them, not standardized form of recording, unreliable; (c) employees knowledge database (WP) - data collected by individual employees, recorded in their notebooks or even unregistered, usually withheld to other employees without a clear order.

The analysis of the information needs defined 40 positions that require registration in the knowledge database used for risk assessment performance. The evaluation of the currently operated data reporting system showed that: (a) in the case of 10 data items, the necessary information are not collected currently, (b) in the case of 18 items, the data collection is insufficient from the point of view of the risk assessment performance, (c) only in the case of 12 items, the data collection is satisfactory from the managers' point of view.

At the same time, it was found that the current form of the data collected is unacceptable from the point of view of analytical work performance. The complete knowledge database (BWS) is only valid in the case of data, the scope of which is satisfactory. They are at the same time the only data that meet the standard required in the risk assessment process. However, even in the case of this group of recorded data, there are 2 data positions that are occasionally entered into the system and they are unreliable. In the case of 2 other data positions, the data are collected only for the needs

Table 2. The main data required for risk assessment performance in the analysed company

Data	Control level	Nature	The scope	Registration form
Data relating to the drivers				
Participation in a traffic accident as a perpetrator	2	I	2	WP
The number of freight transport performed with exceeded working time	3	I	3	BWS
The number of assaults on drivers	1	I	2	WP
Changing the legal regulations for the settlement of the driver	1	NI	3	WP
Work Experience/km driven	2	I	2	WWS
Negative behaviours at work	2	NI	1	–
Owned permissions/completed training	3	NI	2	WWS
Share of the theft being attributable to the carriage performed by the supplier	2	I	1	WP
The number of traffic violations per driver	2	I	2	WP
Data relating to the freight forwarders				
Negative behaviours at work	2	NI	1	–
The share of orders outsourced to the freight forwarders via the stock exchange	2	I	2	WWS
The number unhandled orders	3	I	1	–
The number of incorrectly issued documents	3	I	2	WWS
Past trainings (history)	3	NI	2	WWS
The number of complaints filed by the customer	2	I	2	WWS
Data relating to the tenders				
Negative behaviours at work and in contact with customers	2	NI	1	–
The number of lost customers due to bad communication	3	I	1	–
Negative behaviours at work and in contact with customers	2	I	2	WWS
Data relating to the subcontractors of transport services				
Participation in a traffic accident as a perpetrator	1	I	1	–
The share of incorrectly completed transport processes attributable to the carrier	2	I	2	WWS
The number of orders handled for the company	3	I	3	BWS
The scope of an insurance policy	2	NI	3	WWS
Opinion on the transport market	1	NI	2	WP
Failure to timely provide information about any disruption	1	I	2	WP

(*continued*)

Table 2. (*continued*)

Data	Control level	Nature	The scope	Registration form
Data relating to customers				
Exceeding the maximum weight load	2	I	1	–
The delay in payments regulation	1	I	3	BWS
Negative behaviours associated with the preparation of cargo	1	NI	1	–
Data regarding to the vehicles				
Vehicle electronics failure	2	I	2	WP
Other failures that prevent punctual execution of services	2	I	2	WP
The average maintenance time per vehicle	1	I	1	–
Data relating to the process implementation				
Number of damaged goods	3	I	3	BWS
Number of delayed deliveries	3	I	2	WWS
Number of accelerated deliveries	3	I	2	WWS
Changes in legal regulations	1	NI	3	WP
The number of unrealized freight transports	2	I	3	BWS
The number of thefts	1	I	3	WWS
The number of burglaries to vehicles by immigrants	1	I	2	WP
Data relating to the finance				
Average costs per km on a given route	2	I	3	BWS
The average profit margin on the load of the product group	3	I	3	BWS
The amount of the applicable penalties for improper execution of orders	1	I	3	BWS

of the individual employee (no registration in the system). While incomplete data (called at level 2) are the knowledge of individual employees and in their case, information that are already registered in the system do not keep the required quality and reliability level.

5 Summary

The evaluation of the current data reporting system in the audited company has proven that the current scope and format of the gathered data do not meet the standards required for a knowledge base created for the purpose of risk assessment process performance. The company, in order to implement the concept of risk management for their operational business firstly is forced to make improvements in the current reporting data process. Defined by the managers the data that corresponds to their information needs, must be strictly recorded in the system supporting the activities of the company. The first stage has already been completed. The required data were organized and there was assigned the responsibility for their gathering process to the various organizational units. The evaluation of the scope and form of the gathered data showed gaps which currently exist and require immediate supplement. For this purpose, it is necessary to develop reporting procedures and links them with the employees' incentive system. Only such a labour organization will provide the validity and reliability of the data entered into the system.

The results presented in this paper are the part of a research conducted by the authors and connected with the development of risk management model dedicated to road transport companies. The authors' further research works will be focused on the adaptation of TMS systems to the needs of risk assessment processes performed in the transport companies.

References

1. Abbass, W., Baina, A., Bellafkih, M.: Survey on information system security risk management alignment. In: Proceedings of 2016 International Conference on Information Technology for Organizations Development (IT4OD), 30 March–1 April 2016. IEEE (2016). doi:10.1109/IT4OD.2016.7479260
2. Abkowitz, M.D., Camp, J.S.: An application of enterprise risk management in the marine transportation industry. WIT Trans. Built Environ. **119**, 221–232 (2011)
3. Anderson, E.L.: Scientific trends in risk assessment re-search. Toxicol. Ind. Health **5**(5), 777–790 (1989)
4. Aven, T.: Risk assessment and risk management: review of recent advances on their foundation. Eur. J. Oper. Res. **253**, 1–13 (2016)
5. Aven, T.: Practical implications of the new risk perspectives. Reliab. Eng. Syst. Saf. **115**, 136–145 (2013)
6. Aven, T.: The risk concept – historical and recent development trends. Reliab. Eng. Syst. Saf. **99**, 119–132 (2012)
7. Aven, T.: Perspectives on risk in a decision-making context – review and discussion. Saf. Sci. **47**, 798–806 (2009)
8. Aven, T., Kristensen, V.: Perspectives on risk: review and discussion of the basis for establishing a unified and holistic approach. Reliab. Eng. Syst. Saf. **90**, 1–14 (2005)
9. Aven, T., Krohn, B.S.: A new perspective on how to under-stand, assess and manage risk and the unforeseen. Reliab. Eng. Syst. Saf. **121**, 1–10 (2014)
10. Bajda, A., Laskowski, D., Wrazen, M.: Diagnostics the quality of data transfer in the management of crisis situation. Przegląd Elektrotechniczny **87**(9A), 72–78 (2011)
11. Berle, O., Norstad, I., Asjornslett, B.E.: Optimization, risk assessment and resilience in LNG transportation systems. Supply Chain Manage. Int. J. **18**(3), 253–264 (2013)
12. Commission Staff Working Document on Transport Security. EU Brussels, SWD (2012)
13. COSO Enterprise Risk Management – Integrated Framework. COSO (2004)
14. Cybersecurity considerations for public transit. Recommended Practice APTA SS-ECS-RP-001-14. American Public Transportation Association (2013). www.apta.com/resources/standards/Documents/APTA%20SS-ECS-RP-001-14%20RP.pdf, Accessed Jan 2017

15. The Definitive guide on Risk Management Information system. Aon eSolutions. http://cdn2.hubspot.net/hub/208738/file-30267257-pdf/downloadable_content/The-Definitive-Guide-to-RMIS.pdf, Accessed Jan 2017
16. Ennouri, W.: Risks management: new literature review. Polish J. Manage. Stud. **8**, 288–297 (2013)
17. Gibson, M.: Information systems for risk management. FRB International Finance Discussion Paper No. 585 (July 1997). http://dx.doi.org/10.2139/ssrn.231755, Accessed Jan 2017
18. Gibson, M.S.: The implications of risk management information systems for the organization of financial firms. International Finance Discussion Paper No. 632 (December 1998). http://dx.doi.org/10.2139/ssrn.146910, Accessed Jan 2017
19. Goerlandt, F., Montewka, J.: Maritime transportation risk analysis: review and analysis in light of some foundational issues. Reliab. Eng. Syst. Saf. **138**, 115–134 (2015)
20. Grabara, J., Kolcun, M., Kot, S.: he role of information systems in transport logistics. Int. J. Educ. Res. **2**(2), 1–8 (2014)
21. Haimes, Y.Y., Lambert, J. H., Kaplan, S., Pikus, I., Leung, F.: A risk assessment methodology for critical transportation infrastructure. Final contract report no. FHWA/VTRC 02-CR5. Virginia Transportation Research Council (2002)
22. Heckmann, I., Comes, T., Nickel, S.: A critical review on supply chain risk – definition, measure and modelling. Omega **52**, 119–132 (2015)
23. Heimbecher, F., Kaundinya I.: Protection of vulnerable infrastructures in a road transport network. In: Transport Research Arena Europe, Brussels, pp. 1–10 (2010)
24. Kierzkowski, A., Kisiel, T.: A model of check-in system management to reduce the security checkpoint variability. Simul. Model. Pract. Theor. (2017). doi:10.1016/j.simpat.2017.03.002
25. Kierzkowski, A.: Method for management of an airport security control system. In: Proceedings of the Institution of Civil Engineers - Transport (2016). http://dx.doi.org/10.1680/jtran.16.00036
26. Kisiel, T., Valis, D., Zak, L.: Application of regression function - two areas for technical system operation assessment. In: CLC 2013: Carpathian Logistics Congress - Congress Proceedings, pp. 500–505 (2013)
27. Lubkowski, P., Laskowski, D.: Selected issues of reliable identification of object in transport systems using video monitoring services. Telematics – Support for transport. Commun. Comput. Inf. Sci. **471**, 59–68 (2014)
28. Marhavilas, P.K., Koulouriotis, D., Gemeni, V.: Risk analysis and assessment methodologies in the work sites: on a review classification and comparative study of the scientific literature of the period 2000-2009. J. Loss Prev. Process Ind. **24**, 477–523 (2011)
29. Młyńczak, M., Nowakowski, T., Valis, D.: How to manage risks? The normative approach (in Polish). Problemy Eksploatacji **1**, 137–147 (2011)
30. Patterson, T.: The use of information technology in risk management. White paper (2015). https://www.aicpa.org/InterestAreas/FRC/AssuranceAdvisoryServices/DownloadableDocuments/ASEC_Whitepapers/Risk_Technology.pdf, Accessed Jan 2017
31. PN-ISO 31000:2010: Risk management – principles and guide-lines. PKN, Warsaw (2010)
32. Restel, F.J.: The Markov reliability and safety model of the railway transportation system. Safety and Reliability: Methodology and Applications. In: Proceedings of the European Safety and Reliability Conference, ESREL 2014, pp. 303–311. Taylor and Francis (2015)
33. Risk and safety in the transport sector RISIT – A state-of-the-art review of current knowledge. Research Report, The Research Council of Norway. http://www.forskningsradet.no/csstorage/vedlegg/english_report.pdf, Accessed Jan 2017

34. RiskConsole Risk Management Information System. Mitsui Sumitomo Insurance Group. http://www.msigusa.com/risk_management/docs/RiskConsoleMarketingBrochure2013.pdf. Accessed Jan 2017
35. Roadmap to secure control systems in the transportation sector. Technical Report (2012). https://ics-cert.us-cert.gov/sites/default/files/documents/TransportationRoadmap20120831.pdf, Accessed Jan 2017
36. Rodion, Z.: Analysis of information risk management methods. Bachelor's Thesis. University of Jyvaskyla (2014)
37. Stoneburner, G., Goguen, A., Feringa, A.: Risk management guide for information technology systems. Technical Report no. SP 800–30. National Institute of Standards & Technology Gaithersburg, MD, United States (2002)
38. Szczepański, M.: Insurances in Logistics (in Polish). Poznan University of Technology Publ. House, Poznan (2011)
39. Świeboda, J.: System resilience at an intermodal transhipment node. In: CLC 2015: Carpathian Logistics Congress – Conference Proceedings, Tanger Ltd., pp. 502–511 (2015)
40. Thomas, M.J.W.: A systematic review of the effectiveness of safety management systems. ATSB Transport Safety Report, Australian Government (2012). https://www.atsb.gov.au/media/4053559/xr2011002_final.pdf, Accessed Jan 2017
41. Tubis, A., Werbińska-Wojciechowska, S.: Operational risk assessment in road passenger transport companies performing at Polish market. Article prepared for conference European Safety and Reliability ESREL 2017, 18–22 June, Portoroz, Slovenia (2017)
42. Tubis, A., Werbińska-Wojciechowska, S.: Risk assessment issues in the process of freight transport performance. Article Accepted for Publication in Journal of Konbin (in Press)
43. www.project-seron.eu, Accessed Mar 2017
44. Zajac, M., Swieboda, J.: Process hazard analysis of the selected process in intermodal transport. In: 2015 International Conference on Military Technologies (ICMT), pp. 1–7. IEEE (2015). doi:10.1109/MILTECHS.2015.7153698
45. Zurek, J., Smalko, Z., Zieja, M.: Methods applied to identify causes of air events. In: Reliability, Risk and Safety: Theory and Applications - Proceedings of European Safety and Reliability Conference ESREL 2009, 07–10 September 2009, pp. 1817–1822. Taylor and Francis, Prague (2010)

Language Processing Modelling Notation – Orchestration of NLP Microservices

Tomasz Walkowiak(✉)

Faculty of Electronics, Wroclaw University of Science and Technology,
Wybrzeże Wyspiańskiego 27, 50-320 Wrocław, Poland
tomasz.walkowiak@pwr.edu.pl

Abstract. The paper presents Language Processing Modelling Notation (LPMN). It is a formal language used to orchestrate a set of NLP microservices. The LPMN allows modeling and running complex workflows of language and machine learning tools. The scalability of the solution was achieved by a usage of message-oriented middleware. LPMN is used for developing text mining application with web-based interface and performing research experiments that requires a usage of NLP and machine learning tools.

Keywords: Natural language processing · Text mining · Microservices · Orchestration · Web-based application

1 Introduction

Text mining [1] methods are nowadays available in many software packages. However, building its own computer-based text analysis workflow is a not a simple tasks, especially for non-English languages. Natural language processing (NLP) and machine learning (ML) tools are being developed in different technologies (like Python, Java, R, C++ or OCaml). They are using different formats so their integration is not simple. These problems are overcome by making linguistic tools available in Internet [13] and developing web based applications, which allows to process text data online [4, 7]. However, available solutions are mostly limited to NLP tools and are lacking machine learning methods that are an important part of text mining. Such frameworks were not designed for effectively processing of large corpora. They add a large time overhead due to data transmission and exhaustive data formats [12]. Moreover, they are not aware of possible parallelism in text mining processes and therefore have a poor scalability.

We propose to look on the text mining application a set of "cohesive, independent processes interacting via messages" [3]. It is a definition of microservices [14], an architecture style following service-oriented [2] ideas that has recently started gaining big popularity. Having NLP tools as a set of microservices there is a need to describe cooperation of them to realize specific text mining tasks. There are two approaches to establish such cooperation, known from Service Oriented Architecture (SOA) [2]: orchestration [6] and choreography [9]. Orchestration is based on a central service that sends requests to other services and controls the process by receiving responses.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_44

Choreography, on the other hand, assumes no centralization mechanisms for collaboration. In SOA: these concept where supported by languages as WS-BPEL[1] and WS-CDL[2]. Both of them are a general purpose, based on XML and therefore are very exhaustive. WS-BPEL gained popularity whereas WS-CDL is not commonly used.

We developed a much simpler, human readable (not XML based) orchestration language, dedicated to text mining tasks: Language Processing Modelling Notation (LPMN). It is a formal language defined in LL(*) gramma [8]. Moreover, we implemented the LPMN engine that is capable to orchestrate the NLP microservices providing scalability capabilities.

The paper is structured as follows. We start with identification of requirements for NLP tools orchestration. Next, Language Processing Modelling Notation is presented. It is followed by a description of the LPMN engine and summary.

2 Requirements for NLP Microservices Orchestration

The tools used in processing texts are being developed in different languages (Java, C++, Python and R) and are available in a form of source code or executive binaries. Moreover, many of them (like taggers [10] or name entity recognizers [5]) have large models. Therefore, the time of loading a model is much longer than processing a single text file. The solution is too run a tool as a service with loaded data (models) in memory. Each service running its own process. The usage of services communicating with others by lightweight mechanisms solves also a problem of variety of languages used by NLP and ML tools since there is no need for tight integration.

The tools used in text mining have a common structure [12]. In most cases, they have one input file (or directory), set of configuration parameters (for example defining the used model) and produces one file (or directory) as output.

Let us analyze an example workflow for one of text mining text [11], i.e. clustering. As an input, we have a set of texts that we want to process by a set of NLP and ML tools (implemented as microservices). At first, documents has to be converted to a uniform text format. Next, each text, preferably in parallel, is analyzed by a part-of-speech tagger [10]. Name entity recognizer [5] follows the processing. After it, the features are extracted (like bag of words). Next, the data received from the feature extraction for each input file has to be unified, filtered and weighted. Next, the clustering is performed. In consist of a workflow that is similar to one presented in Fig. 4. The number of parallel run tools is limited by hardware (size of memory, number of processed). That is why a queening system is required to perform such tasks effectively.

The queues has to collect tasks for each type of NLP microservices. Each NLP microservice collects tasks from a given queue and sends back results (or rather messages where results are available) to the orchestrator. Such solution allows providing effective scalability capabilities. The required, mostly used, NLP microservices have to been run in several instances since the queuing systems acts as a load balancer.

[1] http://docs.oasisopen.org/wsbpel/2.0/wsbpel-v2.0.html.

[2] https://www.w3.org/TR/ws-cdl-10/.

3 Language Processing Modelling Notation

3.1 LPMN Statements

The LPMN is a formal language defined in LL(*) gramma [8]. It is structured as a sequence of instructions, each written in a single line of text. The lines are separated by newline characters. All white characters, including additional newlines are ignored by the grammar. The language is designed to accept two types of instructions: LPMN statements and definitions.

Each LPMN statement (syntax diagram of LPMN statement is presented in Fig. 1) models running of chains of tools for inputs (defined by the input element).

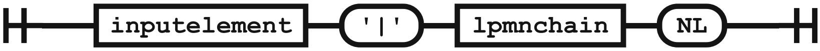


Fig. 1. LPMN statement syntax diagram (The syntax diagrams were generated automatically from LPMN gramma using tool “RRD for ANTLR4” - https://github.com/bkiers/rrd-antlr4)

3.2 Input Elements

Each chain could be run for a single input or for a multiple input. Simple input includes:

- `file`, with one argument: file identifier (it is given by the upload service),
  - for example: `file(/users/default/b87866e7-c28d-43ee-b94b-67ab858c0c3b),`
- `inp`, with one argument: output name, the name refers to any output name (see Sect. 3.3),
  - for example: `inp(“res”).`

Multiple inputs includes:

- `filezip`, with one argument: file identifier (given by the upload service), it is assumed that referenced file is in ZIP format,
  - for example: `filezip(/users/default/a87466e7-c28d-43ee-b94b-67ab858c0c3b),`
- `urlizp`, with one argument: URI, it is assumed that URI points to ZIP file,
  - for example: `urlizp(“http://as.pl/as.zip”),`
- `dspacezip`, with one argument: identifier of corpus in dSpace CLARIN-PL repository[3], identifiers are in the form of persistent identifiers managed by handle.net[4],
  - for example `dspacezip(/11321/319).`

The multiple input allow modeling the parallel processing of all files defined by the input.

[3] https://clarin-pl.eu/dspace/.

[4] http://handle.net/.

3.3 LPMN Chain

The LPMN processing chain consists of a sequence of tool calls or special commands separated by the vertical bar (see Fig. 2). The tool call (Fig. 3) consists of a tool name with optional arguments passed in JSON format[5]. For example following LPMN chain: `any2txt|wcrft2|liner2({“model”:“top9”})` transforms the input data by `any2txt` tool, next by `wcrft2` and finally by `liner2` tool. Whereas the `liner2` tool processing is parametrized by options in JSON format passed within brackets.

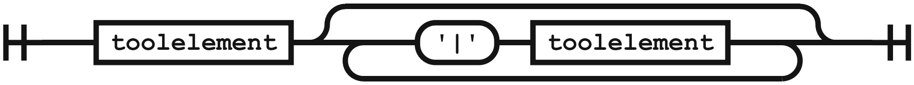


Fig. 2. LPMN chain syntax diagram

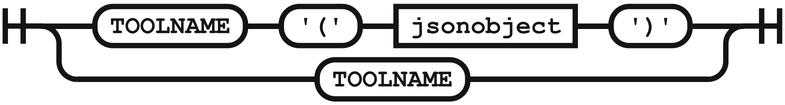


Fig. 3. LPMN tool call syntax diagram

Additionally to tool calls the LPMN chain could contain special commands:

- `out`, with one argument: name, it stores the output file/directory of a preceding tool in a named variable, it could be later used for referencing to intermediate results (see example in Sect. 3.4);
- `div`, with one argument: integer number, divides input text files into smaller parts with a size equal to the passed number (it doesn’t divide words); the div command runs next part of LPMN chain in parallel (like multiple inputs);
- `dir`, the command works as an aggregator, the parallel chain (invoked by multiple inputs or div command) is aggregated into single processing pipe; technical realization is based on coping results of all parallel chains into one directory and running next tool when all parallel chains finished their work.

For example `any2txt|div(1000)|wcrft2|liner2|dir|makezip` will result in a workflow presented in Fig. 4. The input file will be preprocessed by `any2txt` tool. Next, it will be divided in parts of size around 1000 bytes and then each part will be processed in parallel by `wcrft2` and `liner2`. Next, results will be aggregated into one directory and processed once by `makezip`.

[5] http://www.json.org.

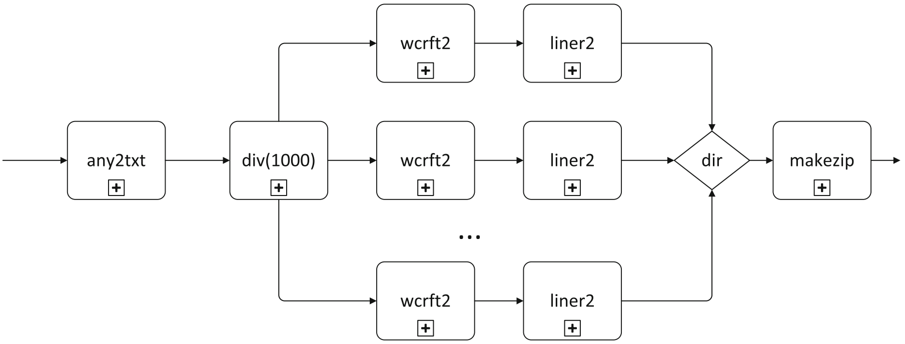


Fig. 4. Workflow of `any2txt|div(1000)|wcrft2|liner2|dir|makezip` statement

3.4 LPMN Definition

For shorting the LPMN code, any LPMN chain could be named and used later. The name prefixed by $ sign could be used in any LPMN statements in a place of tool name (Fig. 6).

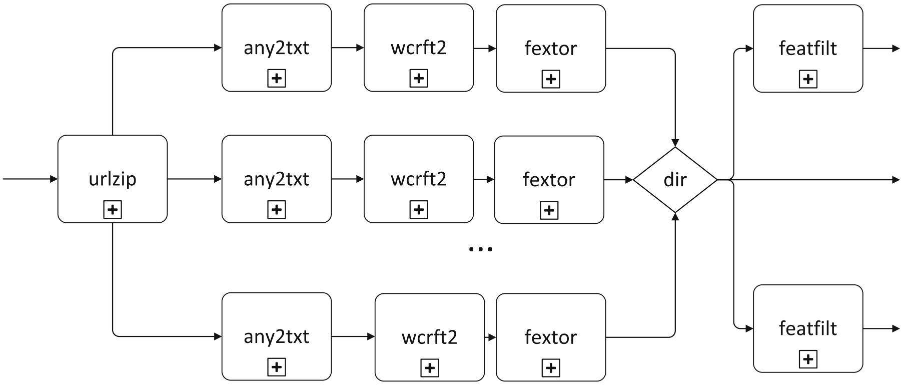


Fig. 5. The workflow of exemplar LPMN code with named outputs and `inp` commands

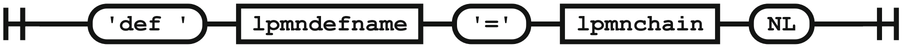


Fig. 6. LPMN definition syntax diagram

Let us analyze following LPMN code:

```
def chain1 = wcrft2|fextor({“features”:“base”})
urlzip(“http://ws.pl/1.zip”)|$chain1|dir|out(“data”)
inp(“data”)|featfilt({“similarity”:“jaccard”})
inp(“data”)|featfilt({“similarity”:“cosine”})
```

The first line defines a chain and named it chain1. Next, it is used in the second line. The above examples illustrates also the usage named outputs. The inp and out statements allows to define to define alternatives for a part of chain. The workflow is presented in Fig. 5. The downloaded files (from ZIP defined by URL) are processed in parallel, and then two separate chains (tool featfilt with different parameters) run. It allows to experiment with different settings of tools without a need to process the corpus repeatedly.

3.5 Parametrized JSON Options

One of assumptions in LPMN development was a fact that tool has one input (depending on previous parts of processing chain) and parameters passed as a constant value. Such approach, even if not very flexible, allowed modeling a large number of different applications (most of application available at http://ws.clarin-pl.eu). The resulting models includes simple chain of tools as well as parallel chains that could be integrated into a single chain (Fig. 4) and next split into parallel chains (Fig. 5).

However, it is not sufficient to model supervised learning in one LPMN code. The supervised learning is a two-stage process. Once the model is created, it is used to classify corpora not seen during learning process using the model obtained during learning. Moreover, there are two different models generated during learning process. First, features (for example bag of words [1]) extracted from texts are weighted. Most of practically used weighting schemas, like for example the most popular in text mining weighting TF-IDF [1], works as transformations. In addition, some of transformation parameters depend not only on features from one text by the entire corpus. Training and testing corpora have to be weighted in the same way, using the same transformation parameters. Moreover, the classification model generated during learning process has to be used during testing. The solution of this problem implemented in LPMN is based on parametrization of JSONs by results of outputs of other tools. In standard JSON gramma,[6] values could be in a form of: strings, numbers, boolean values, null, JSON objects and JSON arrays. We have extended it to special kind of variables, i.e. names of LPMN chain outputs prefixed by $. The exemplar usage of extended JSON notation is shown in Fig. 7. This LPMN code results with a workflow presented in Fig. 8. It allows modeling the process of preprocessing, supervised learning and testing.

```
def ch1=div(1000)|wcrft2|fextor({"features":"base"})|dir
urlzip("http://ws.pl/1.zip")|$ch1|
featfilt({"weighting":"tfidf"})|out("wm")|classify|out("svnm")
urlzip("http://ws.pl/2.zip")|$ch1|
featfilt({"model":$"wm"})|classify({"model":$"svnm"})
```

Fig. 7. LPMN example with parametrized JSONs

[6] http://www.json.org.

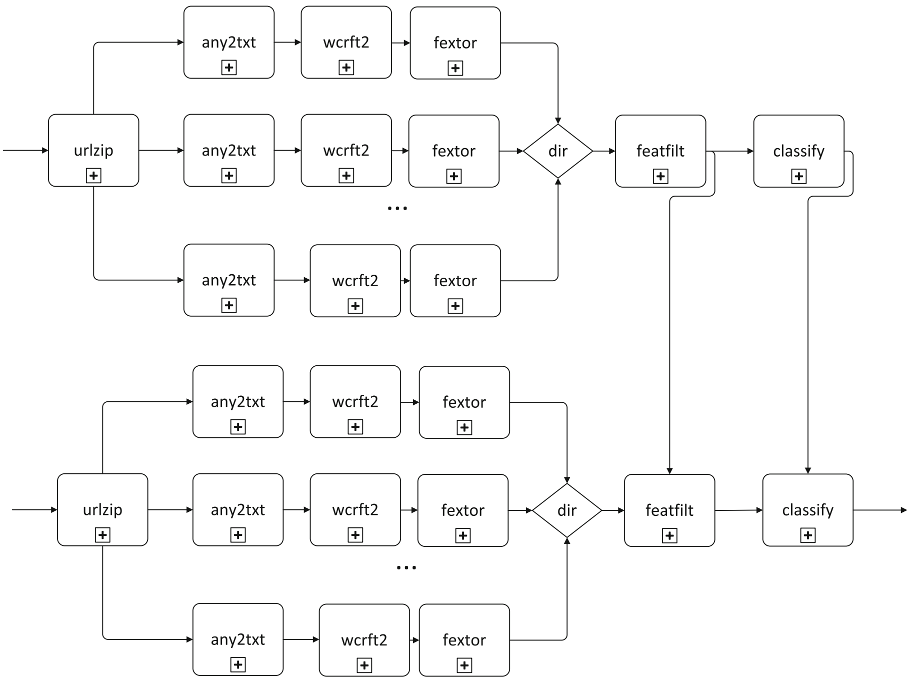


Fig. 8. Workflow of supervised learning and testing (modeled in LPMN as presented in Fig. 7)

4 LPMN Implementation

4.1 NLP Microservices

NLP microservices require a lightweight mechanism that allows sending and receiving messages between language and machine learning tools distributed in the network. We have selected the AMQP[7] protocol and RabbitMQ[8] broker. The AMQP has clients for a large number of platforms (C++, Java, Python) as required by technologies used by language and machine learning tools. Moreover, RabbitMQ has a build in queuing system. Therefore, it fulfils requirements mentioned in Sect 3.1.

Each of tool (microservices) that can be called from LPMN must be implemented as an AMQP consumer. It receives a message that consists of path to input file, path to result file and parameters in JSON format. The paths defined in a message must be accessible (for example by a network shared disk) to a tool service. The aim of the tool is to process an input file (or directory) following passed parameters, put results (file or directory) in an output path and confirm the broker that the task was finished.

Authors developed a set of libraries for Python, C++ and Java that allow a simple deployment of a new tool. It allows running a tool in multiple instances. In case of

[7] https://www.amqp.org/.

[8] https://www.rabbitmq.com.

C++ and Java threads are used. In case of Python processes. The library provides simple abstract class that has to be used by a tool developer. It consists of three methods. Two called during initialization (one called once - static initialization, second for each of running thread/processes) and one called for each incoming message. Such simple solution allows implementing tools that require models to be loaded and instantiated. If the model could be shared among threads/process it should be loaded in static initialization method, if not the second initialization must be used.

There are more than 40 tools available in the current implementation[9]. They include tools developed in such languages like C++, Python, Java, R, Haskell, OCaml and Perl. The number of tools grows instantly. The most used tools includes:

- `any2txt-` converting documents (doc, docx, rtf, pdf, etc.) to UTF-8 texts,
- `makezip-` compressing files (used for shrinking size of result files),
- `convert-` conversion of NLP formats,
- `wcrft2-` tagger for Polish [10],
- `liner2` - name entity recognizers for Polish [5],
- `spacy` - tagger and name entity recognizer for English and German[10],
- `fextor` - tool for feature extracting (calculating of frequency of morphological and grammatical features),
- `featfilt` - tool for feature weighting and calculation of similarities,
- `cluto` - clustering[11] of multidimensional vectors.

4.2 LPMN Engine

The LPMN engine is responsible for running workflows defined by LPMN. The paring of LPMN statements is done by a parser generated by ANTLR framework [8]. It communicates with microservices by AMQP protocol and RabbitMQ broker in the remote procedure call[12] style. The LPMN engine plays a role of a client and a NLP microservice a server. The LPMN engine runs only one thread for each LPMN workflow.

The LPMN engine was deployed at CLARIN-PL[13] infrastructure that forms a private cloud. Software components of the system (microservices, AMQP broker and LPMN engine) are deployed on a set of virtual machines. The scalability is achieved by cloning required virtual machines (with needed NLP and ML tools – microservices) within available hardware resources.

[9] http://ws.clarin-pl.eu.

[10] https://spacy.io.

[11] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview.

[12] https://www.rabbitmq.com/tutorials/tutorial-six-java.html.

[13] http://clarin-pl.eu.

5 Summary

The paper presented the orchestration language, dedicated to text mining tasks: Language Processing Modelling Notation. The language and its implementation allows defining and running flexible workflows of language (like tagger, named entity recognizer, feature extractor) and machine learning tools (like clustering or classification). It is a base for developing Web applications, which process texts online. Each of web application available at http://ws.clarin-pl.eu and many others developed within CLARIN-PL project (for example http://chronopress.clarin-pl.eu/) are using LPMN. Moreover, any user by the REST API[14] could use it. Authors developed a set of exemplar programs in Java, Python and R that allow researchers and developers access LPMN engine and processed texts. The LPMN engine has been used for the last half of a year. During this time more then 500 000 LPMN tasks have been proceeded.

In the next step, we plan to implement the system (LPMN engine and NLP microservices) at university supercomputer center to process huge corpora on demand using available computational resources.

Work financed as part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education.

References

1. Aggarwal, C., Zhai, C. (eds.): Mining Text Data. Springer, US (2012)
2. Bell, M.: SOA Modeling Patterns for Service-Oriented Discovery and Analysis. Wiley, Hoboken (2010)
3. Dragoni, N., Giallorenzo, S., Lluch-Lafuente, A., Mazzara M., Montesi F., Mustafin R., Safina L.: Microservices: yesterday, today, and tomorrow. CoRR, vol. abs/1606.04036 (2016)
4. Hinrichs, M., Zastrow, T., Hinrichs, E.: WebLicht: web-based LRT services in a distributed eScience infrastructure. In: Proceedings of the International Conference on Language Resources and Evaluation, pp. 489–493. European Language Resources Association (2010)
5. Marcinczuk, M., Kocon, J., Janicki, M.: Liner2 - a customizable framework for proper names recognition for polish. Stud. Comput. Intell. **467**, 231–253 (2013)
6. Mazzara, M., Govoni. S.: A case study of web services orchestration. In: A Case Study of Web Services Orchestration, pp. 1–16. Springer (2005)
7. Ogrodniczuk, M., Lenart, M.: A multi-purpose online toolset for NLP applications. LNCS, vol. 7934, pp. 392–395. Springer (2013)
8. Parr, T., Fisher, K.: LL(*): the foundation of the ANTLR parser generator. ACM SIGPLAN Not. **46**(6), 425–436 (2011)
9. Peltz, C.: Web services orchestration and choreography. Computer **36**(10), 46–52 (2003)
10. Radziszewski, A.: A tiered CRF tagger for polish, intelligent tools for building a scientific information platform. Stud. Comput. Intell. **467**, 215–230 (2013)

[14] http://nlp.pwr.wroc.pl/redmine/projects/nlprest2/wiki.

11. Walkowiak, T., Piasecki M.: Web-based natural language processing workflows for the research infrastructure in humanities. In: 5th Conference of the Japanese Association for Digital Humanities, JADH 2015, pp. 61–63 (2015)
12. Walkowiak, T.: Asynchronous system for clustering and classifications of texts in polish. In: Proceedings of the Eleventh International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, pp. 529–538 (2016)
13. Wittenburg, P., et al.: Resource and service centres as the backbone for a sustainable service infrastructure. In: Proceedings of the International Conference on Language Resources and Evaluation, pp. 60–63. European Language Resources Association (2010)
14. Wolff, E.: Microservices: Flexible Software Architectures. Addison-Wesley, Boston (2016)

Type Variety Principle and the Algorithm of Strategic Planning of Diversified Portfolio of Electricity Generation Sources

Volodymyr Zaslavskyi and Maya Pasichna(✉)

Taras Shevchenko National University of Kyiv, Volodymyrska Street 64,
Kyiv 01033, Ukraine
zas@unicyb.kiev.ua, maypas@gmail.com

Abstract. The article provides an approach for the electricity generation companies and their electricity generation portfolios (mix), mainly focusing on the European Union (EU). The model, looking at the management and optimization to structure the electricity generation portfolio for electricity providers, is aimed at defining a system to mix different electricity generation technologies, to maximize the value of the portfolio e.g. ensure energy security and minimize its environmental footprint. Given the limitations of resources and unstable energy market, this would help to understand to what degree electricity generating companies are able to achieve their goals and also how external regulators could achieve their goals. Type variety principle lies at the basis of a developed algorithm.

Keywords: Energy portfolio · Optimization · Energy security · Risk management

1 Introduction

The energy system is a strategic sector with an overall impact on the economy, national security and people welfare; the modeling is seen as a simulation of complex systems with so-called High Cost of Failure (HCF) or critical systems [1, 2]. The Term “HCF” means incompatibility of the costs, resulting from the malfunction of complex system, compared with its normal functioning. To ensure the resilience of such systems Types Variety Principle (TVP) is applied.

TVP is an application of different components e.g. systems, technologies, models, methods, software components. These components perform similar function, but are based on the different principles and each of them can solve the task separately. If combined to a system they will improve resilience and exclude possibility of common cause failure and delivers reliable, long-term performance of complex systems.

TVP differs from diversity theory in a way that it doesn’t seek to explain the richness, turnover, distribution and abundance of all the elements influencing the problem, but aims to combine multiple elements into a single functional system.

TVP is applied through development of single- and multi-criteria mathematical models and algorithms, which are used at all stages of complex systems’ life cycle.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_45

As an example, TVP was used within the concept of formation of different configurations of technologies as well as multi-phased technical and investment solutions for water treatment in the system of water management [3, 4]. The solution (through gradually discrete programming) was the generation of alternative processing technologies and selection one of them, which meets the requirements of existing standards considering the limitations of resources.

This study applies TVP within the algorithm for strategic planning of an optimal diversified electricity generation portfolio through combining the elements of top-down and bottom-up modeling, multi-criteria decision making (Analytic Hierarchy Process - AHP) and optimization concepts of modern portfolio theory.

The article presents a new model for developing optimal electricity production portfolio at the leading EU electricity production companies - EDF, Engie, EON, RWE, Enel, Iberdrola, Vattenfall and Fortum. Total share in electricity generation market of these companies was above 50% in EU-27 in 2014 [18]. So far this problem has been covered mainly in terms of certain technologies or individual countries, excluding the specific electricity generating companies and comparison between them [5–9].

Although the use of AHP and Markowitz portfolio theory is not new, its combination to address the optimization of energy portfolio has not been widely covered before.

The findings of the research can be used by electricity generating companies in general and in other countries (including energy sector of Ukraine) and/or regions while developing the energy portfolio, by the governments while delivering energy and environmental strategies.

The paper is organized into three sections.

- In “Materials and Methods” section the algorithm itself, its steps, as well as methods of collecting input data and delivering calculations are described.
- Section “Results” elaborates on the outcomes of each step of the algorithm, results of the calculations and analysis.
- Discussions on the findings of the research are covered in section “Conclusions”.

2 Materials and Methods

A general description of the model. The model of creation of electricity generation portfolio consists of four steps, which are independent and can be applied in parallel (for convenience the steps are numbered).

The first step is to be “bottom up”: the value ranking of the different electricity generation sources against criteria and factors that electricity companies consider while taking decisions on the structure of electricity generation portfolio. The sources considered were responsible for 97% of generated electricity in EU-27 in 2014 [16].

The second step is considered “top-down”: analysis of balancing of the electricity generation portfolio in order to consider the larger planning, e.g. company strategic targets, governmental policies and regulation on electricity generation portfolio.

Third step: analysis of the changes in the electricity generation portfolio of the companies with regards to the plans and business strategies of these companies. This allows understanding whether the portfolio is strategically aligned and does not contain too many risky and low priority projects, what are the resource levels and changes within the companies.

The fourth step: the forecasting of the electricity generation portfolio is performed and analyzed on how it is consistent with the optimal electricity generation portfolio.

General algorithm of creating electricity generation portfolio is at Fig. 1.

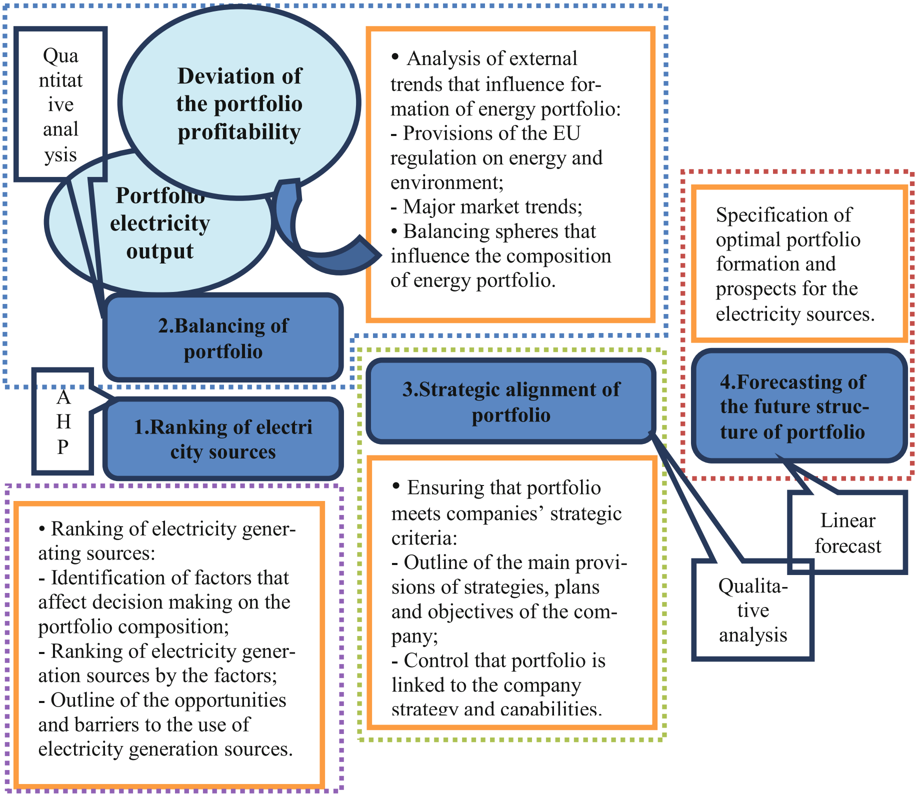

Qua
ntitat
ive
anal
ysis
Deviation of the portfolio profitability
Portfolio electricity output
• Analysis of external trends that influence formation of energy portfolio:
- Provisions of the EU regulation on energy and environment;
- Major market trends;
• Balancing spheres that influence the composition of energy portfolio.
Specification of optimal portfolio formation and prospects for the electricity sources.
2.Balancing of portfolio
A
H
P
1.Ranking of electri city sources
3.Strategic alignment of portfolio
4.Forecasting of the future structure of portfolio
Linear forecast
Qualitative analysis
• Ensuring that portfolio meets companies' strategic criteria:
- Outline of the main provisions of strategies, plans and objectives of the company;
- Control that portfolio is linked to the company strategy and capabilities.
• Ranking of electricity generating sources:
- Identification of factors that affect decision making on the portfolio composition;
- Ranking of electricity generation sources by the factors;
- Outline of the opportunities and barriers to the use of electricity generation sources.

Fig. 1. General algorithm of creating energy generation portfolio

Data collection. The study reviews academic literature and companies' reports (that cover the period 2007 to 2015). Additionally (for AHP purpose), a survey from three strategic experts within different European companies (and different countries) was carried out - semi-restrictive, open-ended interviews were conducted and closed, fixed-response questionnaires were filled in. These experts were senior level managers; they had engineering degree, 15 to 30 years of experience in energy area.

Table 1 below presents which data from the literature, company reports and surveys were necessary for each step of the algorithm and how they were collected.

Table 1. Input data required for each step of the algorithm and their sources

Step No	Input data	Source
1	List of criteria, factors and alternatives for AHP hierarchy	See references [5, 6, 8, 9], Annual, sustainability and financial reports of the companies for 2007–2015 [18–25]
		Interviews with energy experts
	Pair-wise comparison values for AHP	Interviews with energy experts
	RI values	Values from [10]
2	EU regulation on energy and environmental protection	See reference [17]
	Installed capacity of the companies by fuel type for 2007–2014 in main EU countries of operation	Annual, sustainability and financial reports of the companies for 2007–2015 [18–25]
	Weighted LCOE values for new technologies in different countries in 2009–2014	Calculated by authors based on [12–14]
	Standard deviations of the holding period returns (HPR) of the annual costs of technologies	Values from [15]
	O&M correlation coefficients of technologies	Values from [5]
3	Installed capacity of the companies by fuel type for 2007–2014 in main EU countries of operation	Annual, sustainability and financial reports of the companies for 2007–2015 [18–25]
	Targets of the companies with regards to electricity generation activities for the future	
4	Installed capacity of the companies by fuel type for 2007–2014 in main EU countries of operation	

Ranking. Step 1 of the algorithm. AHP application. The AHP model by T. Saaty [10] has been used. The constructed AHP hierarchy (Fig. 2) consists of four levels (top to down): 1. Objective of the hierarchy; 2. Criteria ($F_1 - F_8$); 3. Factors corresponding to criteria ($F_{11} - F_{82}$). 4. Sources of electricity generation - alternatives ($Z_1 - Z_7$).

Given that $\mathrm{Z}_1 - \mathrm{Z}_m$ – a set of alternatives, $m = 7$, consisting of: coal (Z_1), natural gas (Z_2), hydro-energy (Z_3), wind energy (Z_4), solar energy (Z_5), biomass energy (Z_6), and nuclear energy (Z_7). And given that $\mathrm{F}_1 - \mathrm{F}_8$ are eight complex multi-type criteria and $\mathrm{F}_{11} - \mathrm{F}_{13}; \mathrm{F}_{21} - \mathrm{F}_{22}; \mathrm{F}_{31} - \mathrm{F}_{32}; \mathrm{F}_{41} - \mathrm{F}_{43}; \mathrm{F}_{51} - \mathrm{F}_{52}; \mathrm{F}_{61} - \mathrm{F}_{62}; \mathrm{F}_{71} - \mathrm{F}_{73}; \mathrm{F}_{81} - \mathrm{F}_{82}$ are their 19 parent factors then AHP is presented by a set of the following equations – *matrix A of paired-wise comparisons*:

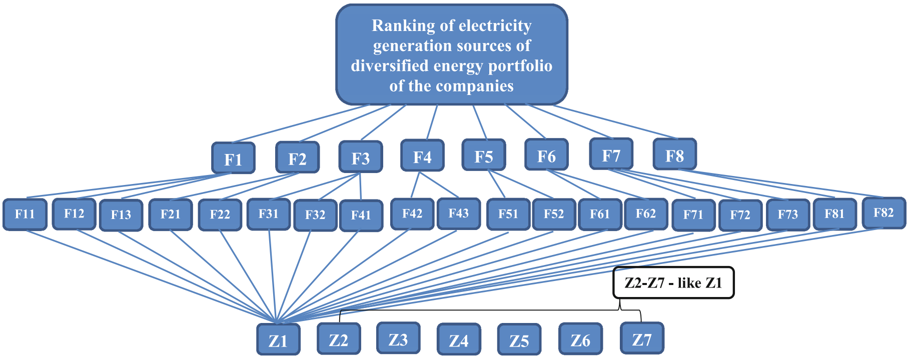


Fig. 2. The multi-level hierarchical structure for evaluating electricity generation sources

$$A^k = \begin{pmatrix} a_{11}^k & \ldots & a_{1n}^k \\ \ldots & \ldots & \ldots \\ a_{n1}^k & \ldots & a_{nn}^k \end{pmatrix}, \tag{1}$$

where k – number of matrix, $k = \overline{1, 28}$;

calculation of vector of local priorities:

$$\mathrm{a}_{\mathrm{i}}^{\mathrm{k}} = \sqrt[\mathrm{n}]{\prod\nolimits_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{a}_{\mathrm{ij}}}, \ \mathrm{i} = \overline{1, \mathrm{n}}, \tag{2}$$

calculation of b_i^k *- normalized vector of* $\mathrm{a}_{\mathrm{i}}^{\mathrm{k}}$:

$$b_i^k = \frac{a_i^k}{\sum_{i=1}^n a_i^k}; \ \sum\nolimits_{i=1}^n b_1^k = 1, \tag{3}$$

calculation of $\lambda_{\max}^{\mathrm{k}}$ *- eigenvalue of matrices:*

$$\lambda_{\max}^{\mathrm{k}} = \sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} \lambda_{\mathrm{i}}^{\mathrm{k}}; \ \lambda_i^k = \sum\nolimits_{j=1}^n a_{ij}^k b_j^k, \tag{4}$$

checking CR - consistency of matrices $(\leq 0,\ 1)$:

$$CI = \frac{\lambda_{\max}^{\mathrm{k}} - n}{n - 1}; \quad CR = \frac{CI}{RI}, \tag{5}$$

where CI- consistency index of the matrix; RI – values of random index of consistency for random matrix of dimension n × n [10].

calculation of b^{Z_m}- global priority vectors of the alternatives:

$$b^{Z_m} = \sum\nolimits_{i=1}^{8} \sum\nolimits_{j=1}^{19} b^{N_0} \cdot b^{N_{F_i}} \cdot b^{N_{F_{ij}}}, \mathrm{m} = \overline{1,7}, \tag{6}$$

where b^{N_0} – normalized vector of priorities of the matrix of judgements against the main objective; $b^{N_{F_i}}$ – normalized vector of priorities of the matrix of judgements of the factors against complex criteria; $b^{N_{F_{ij}}}$ – normalized vector of priorities of the matrix of pair-wise comparison of alternatives against factors.

Balancing. Step 2 of the algorithm. Markowitz theory application. The balancing of portfolios return and risk in accordance to Markowitz theory is outlined by the following key variables [11, 12]: *expected cost of the portfolio:*

$$E_{C_i} = \sum\nolimits_{i=1}^{m} w_i E_{C_i}, \tag{7}$$

expected return of the portfolio:

$$E_{R_p} = \sum\nolimits_{i=1}^{m} w_i E_{R_i}, \tag{8}$$

and standard deviation of the portfolio (σ_ρ) or the total risk of the portfolio

$$E_{(\sigma_p)} = \sqrt{\sum\nolimits_{i=1}^{m} w_i^2 \sigma_i^2 + 2 \sum\nolimits_{i=1}^{m-1} \sum\nolimits_{j=i+1}^{m} w_i w_j \sigma_i \sigma_j \rho_{ij}}, \tag{9}$$

where w_i – share of i source of electricity in the portfolio; E_{C_i} – expected weighted generating cost (LCOE) of i source of electricity per kWh $(\mathrm{C_i})$; m - a set of alternatives; E_{R_i} - expected return of i source of electricity $(\mathrm{R_i})$ (expected return of a physical volume of electricity output per unit cost (a value which is inverse to levelized cost of electricity generation – LCOE [12–14])); σ_i and σ_j – portfolio standard deviations (HPR - holding period returns) in annual costs of the i-th and j-th sources of electricity respectively (Table 2); ρ_{ij} - correlation coefficient between two sources of electricity (takes values from −1 to +1) (Table 3).

Table 2. Standard deviations of HPR of the annual costs of technologies, % [15]

	Construction	Fuel	O&M	CO_2
Z1	23	14	5.4	26
Z2	15	19	10.5	26
Z3	38	0	15.3	0
Z4	5	0	8	0
Z5	5	0	3.4	0
Z6	20	18	10.8	0
Z7	23	24	5.5	0

Table 3. Matrix of correlation coefficients between different electricity generation sources by O&M (operation and maintenance) [5]

	Z1	Z2	Z3	Z4	Z5	Z6	Z7
Z1	1	0.25	0.03	−0.22	−0.39	0.18	0
Z2	0.25	1	−0.04	0	0.05	0.32	0.24
Z3	0.03	−0.04	1	0.29	0.3	−0.18	−0.41
Z4	−0.22	0	0.29	1	0.05	−0.18	−0.07
Z5	−0.39	0.05	0.3	0.05	1	0.25	0.35
Z6	0.18	0.32	−0.18	−0.18	0.25	1	0.65
Z7	0	0.24	−0.41	−0.07	0.35	0.65	1

Strategic alignment. Step 3 of the algorithm. This step is mainly about the resources and the rate of change in electricity generation portfolio structure that is possible and preferable within each company. Therefore it is difficult to use it in an overall model without a deep analysis of each company e.g. current resource levels for different areas and detailed market analysis for each country where they are operational. Therefore has this not been included in the data analysis but is still included in the general method when used by each company.

Forecasting. Step 4 of the algorithm. The linear forecasting of the portfolio was performed based on the trends of the last 10 years [18–25].

3 Results

Step 1. The results of ranking of the electricity generation sources (Fig. 3) indicate that the RES (renewable energy sources) are the highest value of the electricity generation sources for the portfolio within EU for the electricity generation companies today in terms of defined criteria and factors. Nuclear energy and natural gas took the last and penultimate positions respectively.

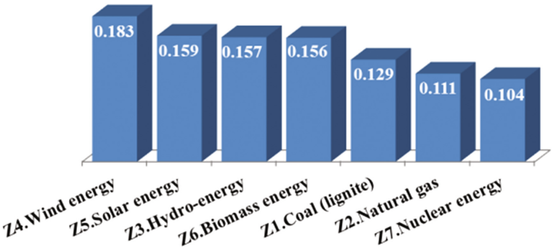


Fig. 3. Weights and ranking of alternatives

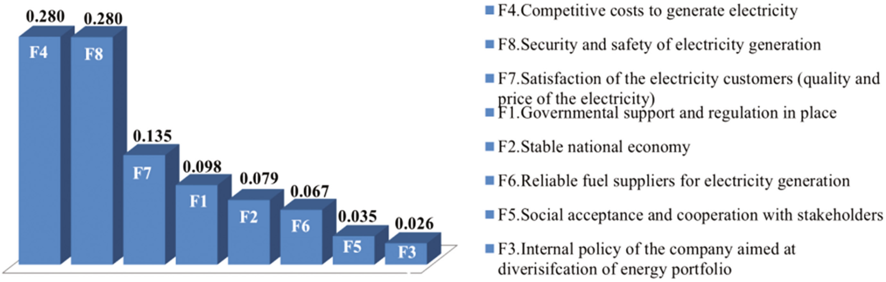


Fig. 4. Weights and ranking of criteria

The cost and safety of electricity generation are the most important criteria in selecting sources of electricity generation. Internal policy of the company is almost three times less important than support from the state and economic stability (Fig. 4).

Step 2. Results of the analysis of portfolio risk and return balancing (Fig. 5) show that companies tend to have higher volume of installed capacity of the portfolios which are characterized by higher risk and higher return. The most risky are Italy, Spain and Germany (mainly due to the predominance of gas generation in Italy and Spain, coal and nuclear generation in Germany, and due to a higher level of LCOE), less risky are France, Sweden, Finland, Belgium, the Netherlands and Portugal.

Taking into consideration the structure of the portfolios, less risky and more economical portfolio (Enel in Spain against Enel in Italy, Fortum in Finland compared to Fortum in Sweden) is not the most diversified portfolio only, but also a portfolio containing a smaller proportion of coal and gas generation (Engie in Germany, Portugal and Spain, E.On in France, Spain and Belgium).

On the other hand, less risky portfolio requires smaller expenditure because of lower value, but means more risk in meeting the required demand.

Thus, despite the high risk of fluctuations in profitability which is characteristic for coal, natural gas, biomass, uranium (Table 3), and the relevant electricity generation portfolios, these fuels tend to have other advantages that are essential for companies in taking decisions about their development – these sources are flexible in terms of transportation, storage, use; predictable in terms of product volume generated to meet demand; the relevant power generating plants have long economic cycle, etc.

In overall, there is no such energy source that can both facilitate and not challenge simultaneously the competitiveness, energy security and environmental impact. Thus, the readiness of the companies regarding the adoption of certain barriers (risks) arises.

Step 3. Degree of strategic alignment of the companies' electricity generation portfolios is low. Despite the transition of the EU companies towards the RES [17] their operations in conventional and nuclear energy remain active. This can be justified by a large production capacity, related infrastructure, current resources (constraints) and dependent customers.

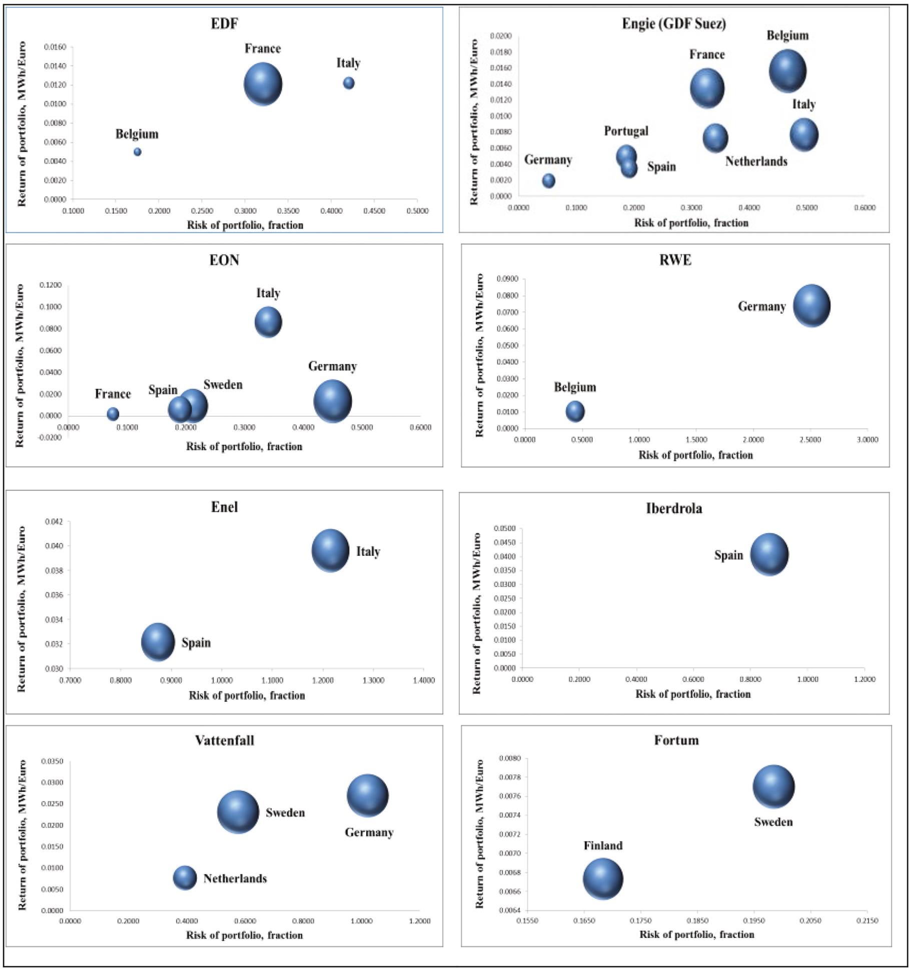


Fig. 5. Balancing of energy portfolios in terms of return (axis Y), risk (X axis) and the installed generating capacity (bubble size) as of 2014

Thus, since 2007 some companies (Enel, E.On, RWE) have significantly reduced "high-risk" facilities that use coal and natural gas due, in particular, to the reduction in electricity demand, on-going economic crisis and RES development. At the same time, these companies remain leaders in these types of installed capacity in the EU.

According to the objectives of the companies, and given the trends of electricity market, the optimal portfolio structure of electricity generating sources is called to be:

- Sustainable in terms of the fuels being available during the next decade(s);
- Foreseeable in terms that the price for energy it generates (wholesale energy prices) is predictable and reflects the costs of generating electricity;
- Being supported by the governmental policies.

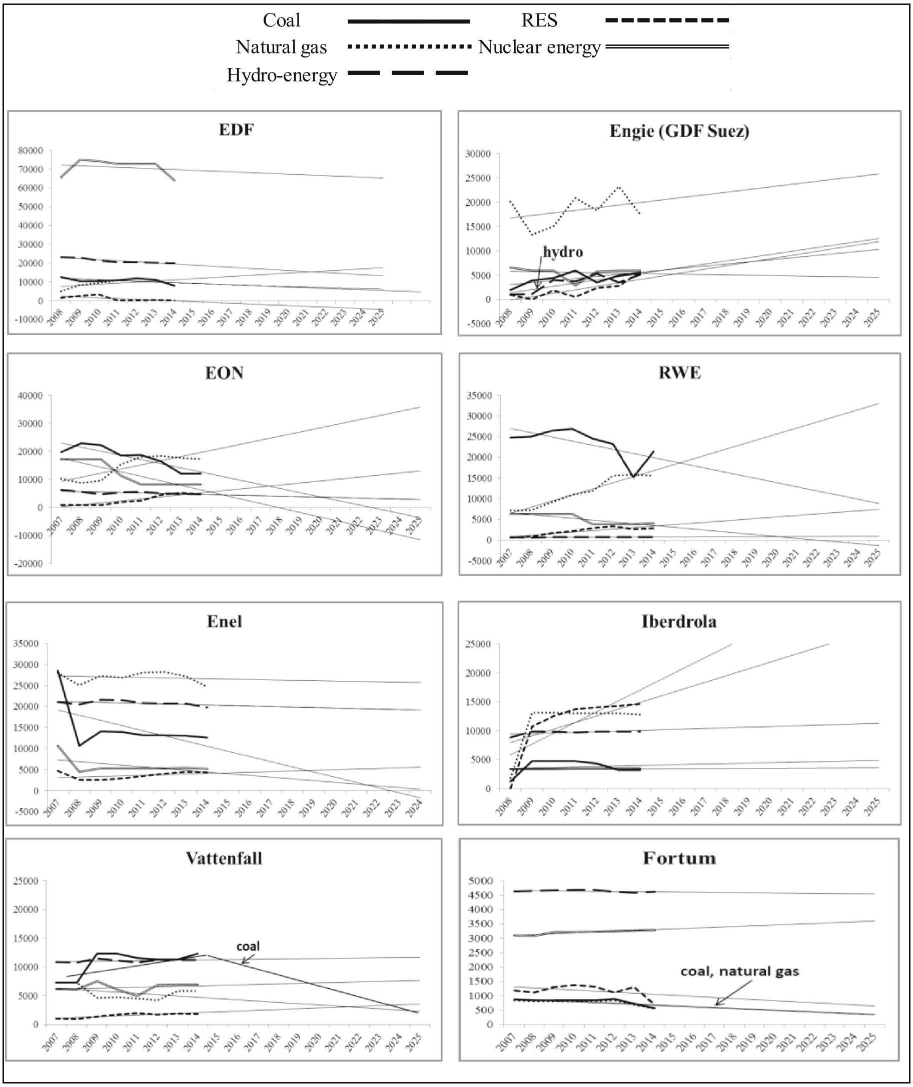


Fig. 6. Past, present and projected composition of electricity generation portfolio, %.

Step 4. Forecasting of the changes in the structure of the energy portfolio of the companies is presented at Fig. 6. It reveals the following:

- Almost all the companies show an increasing trend in installed capacity.
- The downward trend in nuclear capacity is to be compensated by the RES and hydro capacity; natural gas capacity will likely displace the coal generation.

- Linear forecast is simplified and doesn't account for the disruptive behavior of some technologies and political interference (if not taking into account the decision on coal sale of 2016 linear forecast for Vattenfall would suggest increase in coal capacity, which displaces natural gas capacity [23]).

To avoid the rising cost and risk of the electricity generation portfolio these trends can be justified if the companies are ready to replace nuclear generation, on condition of stable natural gas prices, capacities to balance electricity generation from RES.

4 Conclusions

The presented model combines electricity generation sources in a portfolio to account for different aspects of portfolio decision process e.g. strategic goals, constraints of business strategy and political constraints and regulations, technological innovation, and other assumptions on their future directions.

The steps of the model are not done sequentially, but in parallel. They are executed by different types of stakeholders; both within the companies, but some areas are controlled by external stakeholders.

The model shows how all of these stakeholders and types of activities are connected and how they interact. This can therefore be used to both predict the result, and also to affect the future result.

References

1. Zaslavskyi, V.: Type variety principle and peculiarities of study of complex systems with a high cost of failure. Bull. Kyiv Natl Taras Shevchenko Univ. **1**, 136–147 (2006)
2. Zaslavskyi, V., Ievgiienko, Y.: Risk analyses and redundancy for protection of critical infrastructure. Monographs of System Dependability, pp. 161–173. Oficyna Widawnicza Politechniki Wroclawskiej (2010)
3. Zaslavsky, V., Birukov, D., Bekh, O.: Tasks of classification of the levels of regional development. In: Proceedings of the International Conference on Decision Making Under Uncertainty (PDMU-2006), pp. 77–79 (2006)
4. Somlyody, L., Paulsen, C.M.: Cost-effective water quality management strategies in Central and Eastern Europe. IIASA Working Paper WP-92-091 (1992)
5. Awerbuch, S., Spencer, Y.: Efficient electricity generating portfolios for Europe: maximising energy security and climate change mitigation. EIB Papers, vol. 12, no. 2, pp. 8–37 (2007). ISSN 0257-7755
6. Bonin, B., Safa, H., Laureau, A., Merle-Lucotte, E.: MIXOPTIM: a tool for the evaluation and the optimization of the electricity mix in a territory. Eur. Phys. J. Plus. **129**, 1–15 (2014). Springer
7. Taha, R., Daim, T.: Multi-criteria applications in renewable energy analysis, a literature review, pp. 17–30. Springer (2013)
8. Patlitzianas, K., Ntotas, K., Doukas, H., Psarras, J.: Assessing the renewable energy producers' environment in EU accession member states. Energy Convers. Manag. **48**, 890–897 (2007)

9. Cucciella, F., D’Adamo, I., Gastaldi, M.: Modeling optimal investments with portfolio analysis in electricity markets. Energy Educ. Sci. Technol. Part A: Energy Sci. Res. **30**(1), 673–692 (2012)
10. Saaty, T.: The Analytic Hierarchy Process, Planning, Priority Setting, Resource Allocation. McGraw-Hill, New York (1980)
11. Madlener, R., Glensk, B., Westner, G.: Applying Mean-Variance Portfolio Analysis to E. ON’s Power Generation Portfolio in the UK and Sweden. E.ON Energy Research Center, RWTH Aachen University (2009)
12. Salvatore, J.: World Energy Perspective: Cost of Energy Technologies. World Energy Council (2013)
13. Veiga, M., Alvarez, P., Moraleda, M., Kleinsorge, A.: Study on Cost and Business Comparison of Renewable vs. Non-renewable Technologies (RE-COST). Prysma - Calidad y Medio Ambiente S.A. (2013)
14. Ecofys: Subsidies and costs of EU energy. Annex 4–5. https://ec.europa.eu/energy/sites/ener/files/documents/DESNL14583%20Final%20report%20annexes%204%205%20v3.pdf. Accessed 11 Jan 2017
15. Bazilian, M.: Analytical Methods for Energy Diversity and Security: Portfolio Optimization in the Energy Sector. Business & Economics. Elsevier (2009)
16. The Shift Project Data Portal, Energy and Climate Data. www.tsp-data-portal.org
17. EU Energy Markets in 2014. Publications Office of the European Union (2014)
18. RWE Facts & Figures. http://www.rwe.com
19. E.ON Facts & Figures. http://www.eon.com
20. EDF Activity Reports. https://www.edf.fr
21. Fortum Annual Reports. https://www.fortum.com
22. GDF-Suez Environmental Reporting. http://www.engie.com
23. Vattenfall Annual and Sustainability Reports. https://corporate.vattenfall.com
24. Iberdrola Integrated Reports. https://www.iberdrola.com
25. Enel Annual Reports. https://www.enel.com

Author Index

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*,
Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6