Studies in Computational Intelligence 762

Theodor Borangiu
Damien Trentesaux
André Thomas
Olivier Cardin *Editors*

Service Orientation in Holonic and Multi-Agent Manufacturing

Proceedings of SOHOMA 2017

Springer

Studies in Computational Intelligence

Volume 762

Series editor

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

The series “Studies in Computational Intelligence” (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/7092

Theodor Borangiu · Damien Trentesaux
André Thomas · Olivier Cardin
Editors

Service Orientation in Holonic and Multi-Agent Manufacturing

Proceedings of SOHOMA 2017

Springer

Editors
Theodor Borangiu
Faculty of Automatic Control and Computer Science
University Politehnica of Bucharest
Bucharest
Romania

André Thomas
École Nationale Supérieure des Technologies et Industries du Bois
University of Lorraine
Épinal
France

Damien Trentesaux
LAMIH UMR CNRS 8201
University of Valenciennes and Hainaut-Cambresis
Valenciennes
France

Olivier Cardin
Institut de Recherche en Communications et Cybernétique de Nantes
University of Nantes
Nantes
France

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-319-73750-8 ISBN 978-3-319-73751-5 (eBook)
https://doi.org/10.1007/978-3-319-73751-5

Library of Congress Control Number: 2017963010

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 gathers the peer-reviewed papers which were presented at the 7th edition of the International Workshop “Service Orientation in Holonic and Multi-agent Manufacturing—SOHOMA’17” organized on 19–20 October, 2017 by the University of Nantes, France in collaboration with the CIMR Research Centre in Computer Integrated Manufacturing and Robotics of the University Politehnica of Bucharest, Romania, the LAMIH Laboratory of Industrial and Human Automation Control, Mechanical Engineering and Computer Science of the University of Valenciennes and Hainaut-Cambrésis, France and the CRAN Research Centre for Automatic Control, Nancy of the University of Lorraine, France.

The main objective of the SOHOMA’17 Workshop is to foster innovation in smart and sustainable manufacturing and logistics systems and in this context to promote concepts, methods and solutions addressing trends in service orientation of agent-based control technologies with distributed intelligence.

The book is structured in eight parts, each one grouping a number of chapters describing research in actual domains of the digital transformation in manufacturing and trends in future service and computing-oriented manufacturing control: *Part 1*: Advanced Manufacturing Control, *Part 2*: Big Data Management, *Part 3*: Cyber-Physical Production Systems, *Part 4*: Cloud- and Cyber-Physical Systems for Smart and Sustainable Manufacturing, *Part 5*: Simulation for Physical Internet and Intelligent & Sustainable Logistics Systems, *Part 6*: Formal Methods and Advanced Scheduling for Future Industrial Systems, *Part 7*: Applications and Demonstrators, *Part 8*: Production and Logistic Control Systems.

These eight evolution lines have in common concepts, methodologies and implementing frameworks of new **Service-oriented Information and Communication Technologies for intelligent and sustainable manufacturing and logistics systems**.

The theme of the SOHOMA’17 Workshop was “Emerging ICT for intelligent and sustainable manufacturing and logistics systems”.

By defining an Internet-scale platform for networked production encapsulating the right abstractions to link effectively and scalably the various stakeholders of the

Manufacturing Value Chain (materials and component producers, manufacturing plants, technology providers, services, and integrators), the actual vision and initiatives about developing generic architectures and core technologies for the Digital Transformation of Manufacturing (DTM) are presented in the research reported in the workshop, and included in the present book.

An effect of this initiative is shifting from good-dominant logic to service-dominant logic (also expressed as Product-Service Extension) which enhances the utility of product delivered to customer, e.g. installing, configuring/tuning, training the customer, repairing, maintaining/upgrading) adding value in customer operations.

The global vision for DTM refers to: (i) Pervasive instrumenting manufacturing resources, materials flows and environments, (ii) Interconnecting orders, products and resources in a secured Industrial Internet of Things (IIoT), (iii) Taking smart decisions in production management and control by distributing intelligence among multi-agent systems (MAS) acting as information counterparts of physical assets, and orchestrating production workflows as manufacturing services in service-oriented architectures (SOA).

Some issues are addressed in this context:

Markets are currently demanding customized high-quality products with shorter delivery times, forcing companies to adapt their processes by the help of flexible and reconfigurable production plants and supply chains. This leads to the challenge of designing systems that exhibit better reconfigurability, agility, robustness, resilience and responsiveness while ensuring at the same time maintainability, sustainability and long-term performances of their instrumented processes, services, resources, products and logistics systems.

IIoT interconnection is based on the interaction of a multitude of various, instrumented, interconnected and even decision-capable smart objects (generated within the industrial systems), embedded or distant, agentified or purely digital. Developing solutions via these “bottom-up” approaches may lead to emerging behaviour that must be integrated with existing “top-down” approaches, the latter being most of the time centralized or hierarchy-based management systems.

Service orientation of production, control and supervision processes applied to enterprise have gained increasing attention in the recent past, promising a way to create the basis for enterprise agility so that companies deliver new, more flexible business processes that harness the value of service approach from a customer's perspective.

The 7th edition of the SOHOMA workshop focussed on these challenges proposing solutions. It examined the way emerging “bottom-up” behaviours are integrated with “top-down” approaches in an efficient and effective way to constitute hybrid control systems capable to meet industrial needs. It contributed to innovation in sustainable manufacturing promoting concepts, methods, solutions exemplified on case studies addressing trends in the service orientation of agent-based control technologies with distributed intelligence and in the vertical integration of the enterprise.

The scientific event draw on research developed in the last years in the scientific community SOHOMA in the main lines:

- IT-modelling techniques (UML, Object Orientation, Agents, etc.) for automation systems
- Holonic architectures and Multi-agent frameworks for smart, safe and sustainable industrial systems
- Intelligent products, orders and systems in industry
- Service orientation of control and management in the manufacturing value chain
- Issues of agility, flexibility, safety, resilience and reconfigurability in industrial systems with new service-oriented ICT: cloud, Web, SOA
- Cyber-Physical Production Systems and Industrial IoT for Industry 4.0

The application space for these types of development is very broad. The 7th edition of SOHOMA addressed aspects of industrial systems management including classes of applications of the type:

- Manufacturing supply chain: management of production, logistics, transportation, etc
- Industrial services, maintenance and repair operations
- Management and control of discrete and semi-continuous processes with hybrid and supervised control.

The implementing space for these application classes included the new digital transformation frameworks and technologies, as the one advocated by "Industrie 4.0" or "Industry of the future" concepts. Digitalization relates to the interaction between the physical and informational worlds through data and knowledge management as well as virtualization of products, processes and resources managed as services.

Cloud Manufacturing (CMfg), based on the virtualization of control and computing resources, has the potential to move from production-oriented manufacturing processes to customer- and service-oriented manufacturing process networks, e.g. by modelling single manufacturing assets as services in a similar way as SaaS or PaaS software service solutions. In CMfg, all manufacturing resources and abilities for the manufacturing life cycle can be provided in different service models. The Industrial IoT (IIoT) integrated in the cloud allows creating novel network architectures seamlessly integrating smart connected objects, and distinct cloud service providers. The IIoT represents a core enabler for product-centric control and increasing servitization: the product directly requests processing, assembly and materials handling from available providers while it is in execution and delivery. The product monitors its own status, notifies the user when something goes wrong, helps the user to find and access the necessary product-related models and information from manufacturers in the CMfg ecosystem and eases the synchronization of product-related data and models.

To achieve high levels of productivity growth and agility to market changes, manufacturers will need to leverage Big Data sets to drive efficiency across the networked enterprise. A number of contributions formulate proposals for frameworks allowing the development of Cyber-Physical Production Systems (CPPS) with sensing, actuating, computing, communication, control and supervising devices embedded in physical objects, interconnected through several types of networks including the Internet, and providing new business models: Direct Digital Manufacturing, Cloud MES, SOA, product-driven automation, product-service extension for a wide range of industrial applications. CPPS include capabilities for complex event processing and Big Data analytics within the Contextual Enterprise.

Some papers are related to the Industry 4.0 vision to support processes connected with the industrial revolution based on CPPS: pervasive instrumenting, resource virtualization, enterprise integration and networking, Big Data analytics and digital engineering.

A brief description of the book chapters is as follows.

Part I reports recent advances and ongoing research in *Advanced Manufacturing Control*. The contributions point at: MAS architectures for zero defect multi-stage manufacturing based on distributed data collection and balancing of data analysis for monitoring and adaptation among cloud and edge layers to enable earlier detection of process and product variability and knowledge generation by correlating the aggregated data; Supply Chain Management with decision support for multicriteria supplier selection; Environmental assessment using lean-based tools that measure how environmentally efficient manufacturing operations are executed; Operational resilience as the ability of an industrial operation to respond and recover in the face of unexpected or uncontrollable disruptions, and influence the decision-making processes for production control; MAS-based intelligent manufacturing control for enterprises facing high rework rate.

Part II includes papers devoted to *Big Data Management*. The included papers refer to: Distributing data storage by uniformly integrating large number of micro-sensor nodes in objects with cluster-based communication protocol in WSN; Data management architectures for the improvement of availability and maintainability of fleets of complex transportation systems; Solutions to transform raw big data into high-level knowledge usable by engineers and managers, applied in the train transportation domain; Situation awareness in product lifecycle information systems (PLIM) treating massive data, generated by IoT infrastructure.

Part III discusses *Cyber-Physical Production Systems*, used to realize Industry 4.0 compliant systems, and integrating several emergent technologies such as Internet of Things, Big Data, Cloud Computing and multi-agent systems. The following subjects are developed: Agent-oriented software engineering for the requirements analysis, design and validation of CPPS; Self-organized cyber-physical conveyor systems built from intelligent transfer modules; Hybrid approaches to industrial symbiosis through Agent-Based Modelling (ABM) and System Dynamics (SD); The new concept of cooperative safety bubble, aiming at insuring the operator's safety by cooperation among safe robotized systems and safety devices.

Part IV includes recent research in the area of *Cloud- and Cyber-Physical Systems for Smart and Sustainable Manufacturing*. The contributions describe: Replication techniques for resource agents on the cloud, adding the capability of high availability service offered to order agents requesting operations; Relationships between Cloud Computing and Cloud Manufacturing, introducing the Cloud Anything new core-enabling vision based on abstracting low-level resources, beyond computing resources, into a set of core control building blocks providing the grounds on top of which any domain could be "cloudified", including Cloud Manufacturing; Adaptability of holonic manufacturing systems (HMS) obtained by coupling two techniques used in multi-agent systems: resource allocation by Contract Net Protocol and constrained satisfaction by Weak Commitment Search (WCS) algorithm; Software-Defined Networking (SDN) based models and platforms for secure interoperability of manufacturing operations; Real-time machine learning for large-scale manufacturing systems that can predict various scenarios before service degradation occurs, thus allowing for corrective actions (resources capable of reporting their state and a set of KPIs and intelligent products create a high-level loop of information flow, such that real-time data streams flowing from the shop floor into the Manufacturing Service Bus (MSB) are subject to series of map–reduce aggregations, followed by machine learning on aggregated streams; once the information is correlated and consolidated, the MES scheduler uses that information for decision support).

Part V groups papers reporting research in *Simulation for Physical Internet and Intelligent & Sustainable Logistics Systems*. An implementation framework enabling benchmarking of Holonic Manufacturing Systems is proposed; the aim is to develop the design and implementing solution of both the control system and the emulation model of generic, customizable shop floor infrastructure and process classes allowing the standardization of benchmarking initiatives and the application of various benchmarks. Other contributions are related to: Modelling and simulation of complex systems using distributed simulation standard HLA (High Level Architecture) to solve interoperability problems; ORCA hybrid control architecture of railroad PI-hub based on the innovative Physical Internet (PI) paradigm, inspired from the Digital Internet, allows encapsulating goods in modularly dimensioned easy-to-interlock smart PI containers designed to flow efficiently in connected networks of logistics services; Analysis of the robustness of PI-hub cross-docking, a logistics process that consists in receiving freight through unloading docks and then transferring them to the outgoing docks with almost no storage in between; Cyber-Physical Logistics System for Physical Internet, based on the holonic paradigm.

Part VI introduces contributions to *Formal Methods and Advanced Scheduling for Future Industrial Systems*. A new HMI scheduling model is implemented on a real manufacturing scheduling system: the model is adapted to the "groups of permutable jobs" method to cope with shop perturbations. The robustness of production schedules is then evaluated through simulation based on statistical checking; uncertainties related to mass customization (volume and mix of the demand) and the states of the resources (failures, performance degradation, etc.) are considered. A method and technique to design, verify and deploy trusted services for Cyber-Physical Manufacturing Systems, possibly on the fly, is introduced. Discrete event simulation or digital twin is a very effective tool for “what-if” scenarios, for every type of production system. One presented research work transforms the Job-Shop Scheduling Problem (JSSP) into digital twin. The meta-heuristic proposes initial and iteratively improved schedules of orders while the discrete event simulation performs “what-if” scenario for each proposed schedule, thus providing the quality measure of the schedule; this process is repeated until the metaheuristic can no longer provide better schedule. A negotiation scenario using agent-based modelling is finally developed and simulated to deal with dynamic scheduling for disturbed manufacturing processes.

Part VII gathers contributions devoted to *Applications and Demonstrators*. The papers included in this section address aspects of industrial systems management including classes of applications of the type: Measuring the level of similarity of 3D log scans in wood industry based on the iterative closest point method; Supervising daily evaporation in salt marshes using autonomous custom-made devices that embed distance sensors to periodically send measurements of water height to a base station using an Ultra-High Frequency (UHF) radio link; Application of Hybrid Holonic Control Model to gas pipeline network; Deployment of smart AGVs in industry taking into account the tradeoff between smartness and the embedability of their control.

Part VIII is devoted to *Production and Logistic Control Systems*. Papers in this section present new rescheduling heuristic for flexible job-shop problem with machine disruptions; the proposed rescheduling heuristic is based on using routing flexibility to modify the assignment of directly and indirectly affected operations to the others machines. Generic routings for ConWIP sizing in a multi-product environment are then developed in a simulated workshop to select the best generic routing. A study is presented on capacity measurement in two classes of shop floor problems with multifunctional parallel machines. The study emphasizes, in the first example, that the capacity measurement depends not only on machines capabilities, on products requirements and on the imposed working time, but also on the allocation strategy of groups of machines to groups of products, while the second example shows that, in case of machines with unique operation unit time for all operation types, the maximal number of operations executable in given working time is a valid capacity measure. This book section also describes a method of calculating the assembly and delivery plan for groupage cargoes in the special Vehicle Routing Problem with Time Windows (VRPTW) problem of intra-city food delivery.

The studies included in this book show that control paradigms for the manufacturing domain have evolved over time from centralized to decentralized or semi-heterarchical and were mainly driven by the new trends in information and communication technology such as: mobility, connectivity, increase of the decisional capabilities, service orientation and more recently the usage of cloud infrastructures to host control applications, run intensive optimization procedures and store large amount of production and resource data.

The book offers a new integrated vision on *Cloud and HPC, Big Data, Analytics* and *virtualization* in *Computing-oriented Manufacturing*, combining emergent information and communication technologies, service-oriented control of MAS and holonic architectures as well as total enterprise integration solutions based on SOA principles. The Manufacturing CPS (MCPS) philosophy adopts heterarchical and collaborative control as its information system architecture, based on MAS–SOA duality. In this approach, MAS and Data Analysis are the basic technologies proposed to address the requirements and features envisioned by Industrie 4.0, and also taking in consideration the MCPS principles. The first provides the conceptual framework to realize the underline system infrastructure that is required to achieve the desired flexibility and adaptability levels, while the second provides the proper tools capable to analyse and obtain the required information to fulfil the desired system functionalities, also taking advantage of the increased data availability.

The contributions in this book focus especially on how the digital transformation, as the one advocated by "Industrie 4.0" or "Industry of the future" concepts, can lead to improve the maintainability and the sustainability of manufacturing processes, products and logistics. Digital transformation relates to the interaction between the physical and informational worlds. It is realized via virtualization of products, processes and resources managed as services. The theoretical background lies in data and knowledge management, amongst other fields.

All these aspects are treated in the present book, which we hope you will find useful reading.

Bucharest, Romania — Theodor Borangiu
Valenciennes, France — Damien Trentesaux
Épinal, France — André Thomas
Nantes, France — Olivier Cardin
November 2017

Contents

Artificial Intelligence, Autonomous Systems and Robotics: Legal Innovations 1
Raphaël Rault and Damien Trentesaux

Part I Advanced Manufacturing Control

Multi-agent System Architecture for Zero Defect Multi-stage Manufacturing 13
Paulo Leitão, José Barbosa, Carla A. S. Geraldes and João P. Coelho

Multicriteria Supplier Selection for Supply Chain Management 27
Fouzia Ounnar, Olivier Labarthe, Patrick Pujo and Benoit Montreuil

Environmental Assessment Using a Lean Based Tool 41
Andrés Muñoz-Villamizar, Javier Santos, Jairo R. Montoya-Torres and Marta Ormazábal

A Maturity Framework for Operational Resilience and Its Application to Production Control 51
Duncan McFarlane, Rengarajan Srinivasan, Alena Puchkova, Alan Thorne and Alexandra Brintrup

A Case Study of Intelligent Manufacturing Control Based on Multi-agents System to Deal with Batching and Sequencing on Rework Context 63
Emmanuel Zimmermann, Hind Bril El Haouzi, Philippe Thomas, Rémi Pannequin, Mélanie Noyel and André Thomas

Part II Big Data Management

Communicating Aircraft Structure for Solving Black-Box Loss on Ocean Crash 79
Kais Mekki, William Derigent, Eric Rondeau and André Thomas

Data Management Architectures for the Improvement of the Availability and Maintainability of a Fleet of Complex Transportation Systems: A State-of-the-Art Review 93
Damien Trentesaux and Guillaume Branger

Foundation of the Surfer Data Management Architecture and Its Application to Train Transportation 111
Damien Trentesaux and Guillaume Branger

Situation Awareness in Product Lifecycle Information Systems 127
William Derigent and André Thomas

Part III Cyber-Physical Production Systems

Proportional Reliability of Agent-Oriented Software Engineering for the Application of Cyber Physical Production Systems 139
Luis Alberto Cruz Salazar and Hang Li

Empowering a Cyber-Physical System for a Modular Conveyor System with Self-organization 157
José Barbosa, Paulo Leitão and Joy Teixeira

Approaching Industrial Symbiosis Through Agent-Based Modeling and System Dynamics 171
Melissa Demartini, Flavio Tonelli and Filippo Bertani

How to Build a "Cooperative" Safety Bubble for a Reconfigurable Assembly System? 187
Yves Sallez and Thierry Berger

Part IV Cloud- and Cyber-Physical Systems for Smart and Sustainable Manufacturing

Design of High Availability Manufacturing Resource Agents Using JADE Framework and Cloud Replication 201
Silviu Răileanu, Florin Daniel Anton, Theodor Borangiu and Silvia Anton

Reconsidering the Relationship Between Cloud Computing and Cloud Manufacturing 217
Hélène Coullon and Jacques Noyé

On Increasing Adaptability of Holonic Systems 229
Carlos Pascal and Doru Panescu

Software-Defined Networking-Based Models for Secure Interoperability of Manufacturing Operations 243
Radu F. Babiceanu and Remzi Seker

Manufacturing Systems at Scale with Big Data Streaming and Online Machine Learning . . . 253
Octavian Morariu, Cristina Morariu, Theodor Borangiu and Silviu Răileanu

Part V Simulation for Physical Internet and Intelligent and Sustainable Logistics Systems

Proposition of an Implementation Framework Enabling Benchmarking of Holonic Manufacturing Systems . . . 267
Olivier Cardin and Anne L'Anton

Using High Level Architecture in the SEE Project for Industrial Context . . . 281
Simon Gorecki, Gregory Zacharewicz and Nicolas Perry

Control of Rail-Road PI-Hub: The ORCA Hybrid Control Architecture . . . 291
Nhat-Vinh Vo, Thierry Berger, Thérèse Bonte and Yves Sallez

Cyber-Physical Logistics System for Physical Internet . . . 303
Patrick Pujo and Fouzia Ounnar

Simulation for PI-Hub Cross-Docking Robustness . . . 317
Tarik Chargui, Abdelghani Bekrar, Mohamed Reghioui and Damien Trentesaux

Part VI Formal Methods and Advanced Scheduling for Future Industrial Systems

A New HMI Scheduling Model Implemented on a Real Manufacturing Scheduling System . . . 331
Zakaria Yahouni and Nasser Mebarki

Using Statistical-Model-Checking-Based Simulation for Evaluating the Robustness of a Production Schedule . . . 345
Sara Himmiche, Alexis Aubry, Pascale Marangé, Marie Duflot-Kremer and Jean-François Pétin

Trusted Services for Cyber Manufacturing Systems . . . 359
Pascal André and Olivier Cardin

Local Search with Discrete Event Simulation for the Job Shop Scheduling Problem . . . 371
Hugo Zupan, Janez Žerovnik and Niko Herakovič

A Negotiation Scenario Using an Agent-Based Modelling Approach to Deal with Dynamic Scheduling 381
Tsegay Tesfay Mezgebe, Hind Bril El Haouzi, Guillaume Demesure, Rémi Pannequin and André Thomas

Part VII Applications and Demonstrators

Smartness Versus Embeddability: A Tradeoff for the Deployment of Smart AGVs in Industry 395
Guillaume Demesure, Damien Trentesaux, Michael Defoort, Abdelghani Bekrar, Hind Bril, Mohamed Djemaï and André Thomas

H^{2}CM-Based Holonic Modelling of a Gas Pipeline 407
Carlos Indriago, Latéfa Ghomri and Olivier Cardin

An Experimental Device to Supervise the Daily Evaporation in the Guérande's Salt Marshes 421
Antoine Piel, Xavier Huet, Christophe Leray, Benoit Pedrotti, Benoît Parrein and Rémi Lehn

An Iterative Closest Point Method for Measuring the Level of Similarity of 3D Log Scans in Wood Industry 433
Cyrine Selma, Hind Bril El Haouzi, Philippe Thomas, Jonathan Gaudreault and Michael Morin

Part VIII Production and Logistic Control Systems

Generic Routings for ConWip Sizing in a Multi-product Environment 447
Yann Jaegler, Samir Lamouri, Damien Trentesaux, Patrick Burlat and Anicia Jaegler

A New Rescheduling Heuristic for Flexible Job Shop Problem with Machine Disruption 461
Maroua Nouiri, Abdelghani Bekrar, Abderrazak Jemai, Ahmed Chiheb Ammari and Smail Niar

On Capacity Measurement in Two Classes of Shop Floor Problems with Multifunctional Parallel Machines 477
Virginia Ecaterina Oltean, Theodor Borangiu and Silviu Răileanu

The Method of Calculating the Assembly and Delivery Plan for Groups of Cargoes in the Special VRPTW Problem of Intra-city Food Delivery 489
Alexander Lada and Petr Skobelev

Author Index 501

Artificial Intelligence, Autonomous Systems and Robotics: Legal Innovations

Raphaël Rault and Damien Trentesaux

Abstract Ethical, societal and legal issues are rising jointly with the development of autonomous robotic systems immersed into human society. This work focuses on legal aspects and intends to raise the awareness of engineers and researchers in the fields of robotics and artificial intelligence with applications to embedded autonomous systems, cyber-physical systems and self-organizing systems. The paper discusses in detail some recent legal innovations in these fields. Two questions are specifically addressed: how does the lawyer apprehend artificial intelligence and robotics? Which are the existing rules and the necessary legal innovations coming in the next years?

Keywords Artificial intelligence • Data • Robotics • Legal General data protection regulation • Intellectual property • EU civil laws on robotics • Liability • Cyber-physical systems

1 Contextual Elements: Artificial Intelligence and Data

In this section we discuss the main concepts relevant to the topic of artificial intelligence (AI), big data, and machine learning for robot integrated as multiple resources in manufacturing and services. Then, some discussions are initiated from a lawyer's point of view about main expected guarantees regarding research and development in these topics.

The ISO 2382-28 standard defines "artificial intelligence" as the capacity of a functional unit to perform functions generally associated with human intelligence,

---

R. Rault
Alter via Avocats, 7 rue de l'Hôpital Militaire, 59800 Lille, France
e-mail: rrault@alter-via.fr

D. Trentesaux (✉)
LAMIH UMR CNRS 8201, SurferLab, University of Valenciennes,
Le Mont Houy, 59313 Valenciennes Cedex, France
e-mail: damien.trentesaux@univ-valenciennes.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_1

such as reasoning and learning. In principle, no physical envelope is required for the functional unit. Without data, there is no artificial intelligence. We might think that fully autonomous artificial intelligence does not need anything to maintain and gain new skills. On the contrary, the more a machine is autonomous, the more it is actually relying on data [1]. Everybody faces the huge amount of data generated and gathered in daily life or in industry, which has led to the so-called "Big Data" that can be defined as a set of concepts, techniques and practices related to the intensive use of large data sets. Five "V features" characterize big data: volume, variety, velocity, veracity and value. Four phases define the processing of big data: (1) acquisition/enrichment, (2) integration/homogenization, (3) analysis/knowledge extraction, (4) visualization/interpretation [2]. "Machine learning" aims to give AI the ability to learn and improve with experience to make predictions and take decisions. AlphaGo Zero (Google Deep Mind AI) just learned the "GO" Japanese game from scratch and by itself and became world champion in only three days. AI searches correlations between several events but we still need human analysis for the interpretation of results (such as accidental correlation without causal relationship between the number of people drowned after a fall from a fishing ship and the wedding rate in Kentucky between 1999 and 2010).

From a lawyer's point of view, data used by artificial intelligence should guarantee the followings [3]:

- **Equity**: Data should avoid biased results. For instance, the European Union Commission warned Google on July 14th, 2016 for alleged abuse of dominant position by systematically favouring its price comparison service in its results pages and by artificially limiting the possibility for third party websites to display the contextual advertisements emanating from Google competitors.
- **Protection of personal data**: French law prohibits all automatic decisions with legal effects (without human intervention) based on personal data. The European Union General Data Protection Regulation (GDPR) requires an opt-in (prior specific consent) for profiling with legal effects. The profiling is defined as any form of automated processing of personal data consisting of the use of personal data to evaluate certain personal aspects relating to a natural person, in particular to analyse or predict aspects concerning that natural person's performance at work, economic situation, health, personal preferences, interests, reliability, behaviour, location or movements.
- **Transparency**: It is a founding principle of the French data protection law and GDPR (right to access to the logics underlying profiling). The French data protection authority (CNIL) sentenced FACEBOOK on April 27th 2017 to a 150,000 € fine because of a massive crossing of users' personal data not mentioned expressly.
- **Diversity**: Browsers and algorithms tend to favour the popularity of results at the expense of diversity, which narrows the information horizon.
- **Neutrality**: This means the absence of discrimination in access to data and calculation tools. Levers of neutrality can be: user education, public policies

(ex: The French AI initiative and French Open Data initiative) and technical solutions.
- **Accuracy**: GDPR article 5 specifies that personal data must be accurate and, if necessary, kept up to date.
- **Intellectual property**: Data are intangible assets of companies. But who owns source data? Enhanced data coming from the crossing of several databases? New data created by the algorithm?

Since they will handle huge quantity of data, researchers working with AI should then keep in mind these guarantees. For example, a growing number of institutions ask respecting the guarantees above defined for each project in exchange to the subject to grant funding received. From the authors' point of view, the last guarantee is the most challenging one in the context of autonomous learning systems. The next section addresses this aspect.

2 Robotics and Intellectual Property

Until now, intellectual property was hold by humans, but this may change if one considers that robots may generate novel concepts, ideas or even artistic items, whatever the form of this novelty.

The elements defining intelligent robot systems are: hardware, energy source, ability to act in the physical world, ability to perceive its environment, decision-making capability, and learning ability [4]. The secondary elements are autonomy and communication capacities. Robot applications exist in a variety of activity fields: industry, medicine (surgery robots, human repair and condition improvement), services (care and personal robots), and transportation (autonomous vehicles). "Z MACHINE" is a Japanese music band only composed of robots, which create their own songs. "E-David" is a robot which creates artistic paintings and drawings. Robots' components and works can be protected by intellectual property. The designer of *robot's hardware* (plans, schemes, material specifications) can be protected by private copyright or open source copyright (open robot hardware) or designs and models. The programmer of the *robot's software* (original source code) is protected by private copyright or open source copyright. The trademark's holder is protected for certain products and services for the *robot's name*. *Original creations and inventions of the robot* are protected only when there is a human co-author.

When the original creations and inventions only come from the robot, there is no copyright, as there is no legal personality to the robot up to now. The next section addresses such a need to define civil laws for robotic systems.

3 EU Civil Laws on Robotics

On October 18th 2017 the results' summary of the public consultation on robotics and Artificial Intelligence was published by the European Union Parliament [4]. The panel was composed of European citizens and organizations. An overall positive attitude towards robotics and AI was noticed but a need for careful management was also expressed. The main concern is the threat to privacy. 90% of the panellists at EU or international level think it is necessary to regulate development in robotics and AI. The most urgent regulation resides on autonomous vehicles.

The main goals of the February 16th 2017 EU Parliament resolution are: human safety (with the rule "first, do not harm" coming from Asimov Laws), human privacy, human integrity, human dignity, and human autonomy.

An interesting notion has been suggested: the concept of "electronic person". The legal status for an "electronic person" has been proposed by the EU Parliament for Smart Autonomous Robots (SAR): any robot that makes autonomous decisions intelligently or interacts independently with third parties. SAR acquires autonomy through sensors and/or by exchanging data with its environment (interconnectivity), exchanges and analyses data, is self-learning (optional criterion), has a physical support, adapts its behaviours and actions to its environment, and is not alive (in a biological sense). A SAR should be held reliable for its actions and the greater a robot's learning capability or autonomy is, the lower other parties' liability should be; also, the longer a robot's 'education' has lasted, the greater the liability of its 'teacher' should be.

The main recommendations of the EU Parliament are as follow:

1. **Definition and classification of smart robots**: Cyber Physical Systems (CPS), autonomous systems, smart autonomous robots, subcategories.
2. **Registration of smart robots**: This registration would be made to the EU Agency for Robotics and AI which will be created in the future.
3. **Civil law liability**: A mandatory insurance for each robot is required.
4. **Interoperability**: This means access to source code and intellectual property rights to the electronic person.
5. **Charter on robotics**: Includes the design process, ethics review, audit controls, etc.
6. **Code of ethical conduct for robotics engineers**: The principles are: beneficence (in the best interests of humans), non-maleficence (first, do not harm), human autonomy (in the interactions with robots), and justice (affordability). The fundamental rights are: precaution, inclusiveness (transparency), accountability (of robotics engineers on social, environmental and human health impacts of robotics), safety (human well-being), reversibility (undo the last action), privacy (appropriate goals and means), maximizing benefit and minimizing harm (risk assessment and management protocols)
7. **Code for research ethics committees (REC)**: Resides on independence, competence, transparency and accountability, monitoring.

8. **Standard license for designers**: In which designers should protect the users and also follow the rules: trustworthy system design principles, privacy by design, opt-out mechanisms (kill switches), compliance with legal and ethical principles, reconstruction and traceability of robot's actions, predictability of robot's behaviour, analysis of the predictability of human-robot systems failures, tracing tools at design stage, design and evaluation protocols, robots to be identifiable as robots by humans, safeguard safety and health of humans interacting with robots, positive opinion from a REC before an IRL test.
9. **License for users** with rights: The rights are: using of the robot without risk or fear of physical or psychological harm, and expecting the results the robot is designed for. The duties are: taking into account the robot limitations (perception, etc.), respecting: human physical and psychological frailty and human emotional needs (care robots), privacy rights of individuals, prior consent of the individual before processing personal data, not contravening ethical or legal rules, not modifying a robot to enable it to function as a weapon.

An EU Directive is recommended by the EU Parliament in this field (the Directive needs transposal in national laws of each EU country while EU Regulations are directly bindings).

4 Contractual Frame When Developing Artificial Intelligence

The contractual framework to develop artificial intelligence concepts, methods, and applications depends on the phase of the AI lifecycle.

For example, in the "**Build**" stage (**design**), AI mainly lies on software development contract and software integration contract.

In the "**Run**" stage (**use**), AI mainly lies on license (which needs to define the territory, duration, scope of transfer of rights: use, reproduction, distribution, adaptation), maintenance and hosting contracts. Robotics also needs contracts on robots' hardware: design, production, distribution, selling/leasing, maintenance. The main legal/contractual obligations are: security, compliance, prior information, and delivery.

This classical way of considering the contractual framework has to evolve in the context studied in this chapter. For example, if an AI creates something new that cannot be directly linked to the responsibility of the researcher who designed the AI concept or application, and more, if what has been created by AI is used by others, then the AI should be able to license its creation.

This evolution concerns also the legal liabilities, as will be discussed in the following section.

5 Legal Liabilities for Autonomous Systems

As soon as damage occurs because of a failure and if we can prove the causality link between the damage and the failure, a liable person is sought. This liable person can come from the AI expert team (the AI designer or software programmer) or the robot team (the robot designer, supplier, owner, distributor, maintainer) or the user or third parties.

The *existing legal regimes* can be: special liability of the designer and the programmer for defective products (structural design risk but with development risk exemption). But who is liable when we use open source software? Is it the author or the community? It can also be the liability of the user's personal fact, but can the user control the behaviour of the robot? This is similar as for a fully autonomous vehicle: who is the driver? It can also be criminal liability, with the obligation of security of personal data, pre-contractual information, and compliance. Lastly, the intellectual guardianship of the thing leads us to the liability of the guardian (e.g.: alone in a fully autonomous vehicle, are we the driver? the passenger? the user?).

As an illustration of the impact autonomous learning systems may have on society and legal affairs, the next section provides an illustration of what can be proposed in terms of liability using the context of cyber-physical systems (CPS). These considerations represent preliminary work, with no intention to impose a point of view; they can be rather considered as an intention to raise the awareness of the reader on emerging and urgent needs in what could be identified as a new research field dealing with ethical behaviour of autonomous systems.

6 Application to Cyber-Physical Systems

First, the previous evolution of laws and legal rules defined in Sects. 2 and 3 is here illustrated through the evolution of technical systems towards cyber-physical systems, with a focus on the challenging novel research domain dealing with ethical behaviour of CPS when they are sufficiently autonomous. For this purpose, we start from the basics behind these legal aspects which is *dependability* [5], since dependability deals with the fact that systems may fail or behave erratically; this perspective leads to the necessity to establish responsibilities in case of damage or hazardous event.

Dependability is basically constructed on the following indicators that are used in system engineering and safety studies:

- **Reliability**: The ability that a system operates satisfactorily during a determined period of time and in specific conditions of use. It can be expressed as the mean time to failure (MTTF).
- **Availability**: The ability to perform when called upon and in certain surroundings. It can be expressed as a percentage, that is the proportion of time a system is in a functioning condition over a given time period.

- **Maintainability**: Is the capacity of a system to be maintained, where maintenance is represented by the series of actions taken to restore an element to-, or maintain it in an effective operating state. It can be expressed as the mean time to repair (MTTR).
- **Security**: Is the ability to operate satisfactorily despite intentional external aggressions. In that sense, security is related to external events, while reliability is related to internal events.

Given the previous discussion, results the need to establish the status of an "electronic person" applied to autonomous Cyber-Physical Systems, that consists in merging the cyber- and a physical world [6]. For example, one could argue that RAMS should evolve towards RAME studies, where E, which is ethicality, encompasses the S (security). As a consequence, RAME studies add ethics [7].

- **Ethicality**: The capability of a system to behave according to an ethical manner.

In the context of autonomous CPS, the notion of ethicality means:

- Integrity, that is the ability to behave so that the others trust the information coming from the CPS and are confident in the ability of the CPS to engage actions to reach a clear objective, potentially varying, but which must be announced and publicized in that case (ethical design).
- Safety concerns people involved and in direct connection to the CPS. The CPS should behave in a safe way, limiting the risk of injuries to human beings. This part is very complicated. For example, a safe decision for an autonomous high-speed train that detects lately an animal on the track could be to brake to avoid the collision while at the same time, it risks to hurt several of its 1,000 passengers (the ones that are at this moment not seated) with this emergency brake.
- Security: Could security considered even against human behaviour? For example, should the auto pilot of a plane be able to override the command of a human pilot if it may lead to crash the plane (e.g., suicide)? If yes, how to ensure that this override may not lead to a more critical situation (e.g., difficulty for the cruise CPS to land the plane close to an airport full of people)?
- Altruism: means welfare of third parties (other CPS or human beings). Altruism may be contradictory to one or several of the other RAM indicators. For example, an altruist autonomous boat that detects "SOS" messages, delays its current mission to help the senders of the SOS. But through this altruist decision, it will de facto reduce its availability (its mission may be cancelled) and this decision may impact its reliability as well (overuse of its turbines for example). A more critical example is the one of an autonomous car that has to break the law in order to save life (e.g., emergency trip to hospital, line cross to avoid a pedestrian, etc.)
- Accountability: Actions performed by the CPS engaging its liability in case of problem; that is why insurance funds need to be created for injured persons.

Table 1 Ethical and legal responsibilities according to the level of complexity of the designed autonomous systems

<table>
<tr><th>Complexity level</th><th>Characteristics of the level</th><th>Potential ethical and legal responsibilities (cumulative)</th></tr>
<tr><td>Level 1</td><td>Single autonomous physical system
Fully reliable, non learning</td><td>User, owner</td></tr>
<tr><td>Level 2</td><td>Single autonomous physical system
Limited reliability, non learning</td><td>+ Maintainer, supplier manufacturer, integrator</td></tr>
<tr><td>Level 3</td><td>Single autonomous physical system
Limited reliability, learning</td><td rowspan="2">+ Autonomous physical system (s)
Researchers/scientists/PhD students</td></tr>
<tr><td>Level 4</td><td>Multiple autonomous physical systems in a CPS, limited reliability, learning</td></tr>
</table>

Secondly, and following the discussion that took place in Sects. 4 and 5, one could also address the potential ethical and legal responsibilities when designing an autonomous system, as illustrated in Table 1 [8] where the higher the level, the more complex the autonomous system. Level 4 is the one of the CPS where it can be seen that a researcher can be sued because the autonomous CPS he designed caused an accident during its use, a long time after its design.

All these elements can be seen as really futuristic, but industrialists working to design CPS in transportation e.g., autonomous cars, ask themselves these questions. The same occurs in train transportation: the French SNCF has launched a set of projects with the national research institute RAILENIUM to put on track in few years the autonomous train. These questions have rapidly been raised by engineers from the enterprise. Up to now, no answer can be provided by researchers.

7 Conclusion

The current legal system can address a majority of situations but cannot address all the issues raised by the democratization of civil and industrial robotics. Such a new legal system should consider the complexity of robots and go beyond the simple legal framework to also require ethical commitments from the main actors. The solutions should come from the European Union and/or international treaties. But these treaties will rely partially on knowledge yet to generate and scientific research yet to develop.

Acknowledgements This work is done within the context of the project “Droit des robots et autres avatars de l’humain” (robotic laws and other avatars of the human). The authors would like to thank the University of Strasbourg for its support to this project. This research work is also done in the context of the project “Etude de faisabilité de l’intelligence artificielle pour le train autonome” (feasibility study on artificial intelligence for the autonomous train) supported by the Railenium technological research institute and funded by the French railway agency SNCF. The authors would also like to thank these institutions for their supports.

References

1. Delcroix, G.: Des objets connectés aux objets autonomes : quelles libertés dans un monde robotisé ? https://linc.cnil.fr/fr/des-objets-connectes-aux-objets-autonomes-quelles-libertes-dans-un-monde-robotise (2017)
2. Bouzeghoub, M., Mosseri, R.: Les Big Data à découvert, CNRS éditions (2017)
3. Stoyanovich, J., Abiteboul, S., Miklau, G.: Data, responsibly: fairness, neutrality and transparency in data analysis. In: International Conference on Extending Database Technology, 15 Mar 2016
4. Nevejans, N.: Traité de droit et d'éthique de la robotique civile, LEH Edition (2017)
5. Pascual, D.G., Kumar, U.: Maintenance Audits Handbook: A Performance Measurement Framework. CRC Press (2016)
6. Lee, J., Bagheri, B., Kao, H.-A.: A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems, Manufact. Lett. **3**, 18–23 (2015)
7. Trentesaux, D., Rault, R.: Ethical behaviour of autonomous non-military cyber-physical systems. In: XIX International Conference on Complex Systems: Control and Modeling Problems, Samara, Russia, Sept 2017
8. Trentesaux, D., Rault, R.: designing ethical cyber-physical industrial systems. In IFAC-PapersOnLine, pp. 14934–14939 (2017)

Part I
Advanced Manufacturing Control

Multi-agent System Architecture for Zero Defect Multi-stage Manufacturing

Paulo Leitão, José Barbosa, Carla A. S. Geraldes and João P. Coelho

Abstract Multi-stage manufacturing, typical in important industrial sectors, is inherently a complex process. The application of the zero defect manufacturing (ZDM) philosophy, together with recent technological advances in cyber-physical systems (CPS), presents significant challenges and opportunities for the implementation of new methodologies towards the continuous system improvement. This paper introduces the main principles of a multi-agent CPS aiming the application of ZDM in multi-stage production systems, which is being developed under the EU H2020 GO0D MAN project. In particular, this paper describes the MAS architecture that allows the distributed data collection and the balancing of the data analysis for monitoring and adaptation among cloud and edge layers, to enable the earlier detection of process and product variability, and the generation of new optimized knowledge by correlating the aggregated data.

Keywords Cyber-physical systems · Zero defect manufacturing · Multi-agent systems · Multi-stage manufacturing

---

P. Leitão · J. Barbosa (✉) · C. A. S. Geraldes · J. P. Coelho
Polytechnic Institute of Bragança, Campus Sta Apolónia, 5300-253 Bragança, Portugal
e-mail: jbarbosa@ipb.pt

P. Leitão
e-mail: pleitao@ipb.pt

C. A. S. Geraldes
e-mail: carlag@ipb.pt

J. P. Coelho
e-mail: jpcoelho@ipb.pt

P. Leitão
LIACC – Artificial Intelligence and Computer Science Laboratory,
Rua Campo Alegre 1021, 4169-007 Porto, Portugal

J. Barbosa · J. P. Coelho
INESC-TEC, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_2

1 Introduction

Nowadays, the modern and competitive companies must be able to track rapid technological changes while carrying out the manufacture of products with complex features, which commonly requires the assembly of a large number of components. Moreover, the dynamic nature of today's manufacturing environments compels organizations to an incessant reassessment in an effort to respond to continuous challenges in the field of manufacturing management.

Multi-stage manufacturing systems [7], which are typical in important industrial sectors such as automotive, house hold appliance and semiconductor manufacturing, are inherently complex. Among other characteristics, it is common to have multiple stages with mixed sequential and/or parallel configurations, feedback/feedforward loops, and mixed data types that arise from multiples processes. In this context, the application of the zero defect manufacturing (ZDM) philosophy [10], together with recent technological advances in cyber-physical systems (CPS), namely Internet of Things (IoT), big data, and advanced data analytics, presents significant challenges and opportunities to develop new methodologies aiming at continuous improvement of process efficiency and product quality.

In the actual European manufacturing context, the development and transfer of advanced technology for the industry is of strategic importance. This technological contribution aims to boost competitiveness while targeting several aspects of the manufacture processes such as reduce waste and costs, and increase processes efficiency and quality tracking. In this context, the EU H2020 GO0D MAN (aGent Oriented Zero Defect Multi-stage mANufacturing) project (see http://go0dman-project.eu/) aims at integrating and combining process and quality control for multi-stage manufacturing systems using a distributed system architecture built upon an agent-based CPS and smart inspection tools designed to support ZDM strategies.

In this approach, the multi-agent system (MAS) [11] infrastructure, combined with data analytics, provides real-time and early identification of deviations allowing to prevent the occurrence of defects at a single stage and their propagation to downstream processes, enabling the global system to be predictive by early detecting faults, and proactive by self-adapting to different conditions. This approach is aligned with Industry 4.0 trends, contributing for achieving a dynamic and continuous system improvement in multi-stage manufacturing environments, addressed by the ZDM philosophy.

The objective of this paper is to introduce the main principles of the designed agent-based CPS to support zero defects multi-stage manufacturing. The proposed architecture is based on the previous successful EU FP7 GRACE project [5], adapted to handle the idiosyncrasies introduced by multi-stage manufacturing systems and ZDM strategies, providing an infrastructure for the distributed intelligence and digitization.

The rest of the paper is organized as follows: Sect. 2 overviews the challenges of applying ZDM in multi-stage manufacturing systems and Sect. 3 describes the agent-based CPS architecture to implement ZDM strategies. Section 4 describes the distribution of data analysis by cloud and edge layers and Sect. 5 presents some interaction patterns to implement the ZDM strategies. Finally, Sect. 6 rounds up the paper with the fundamental concluding remarks and points out some directions for future work.

2 Challenges of Applying ZDM in Multi-stage Manufacturing

The ZDM philosophy aims to achieve a steady-state production with no defects whatsoever at the end of the manufacturing process. This concept encompasses a large number of mechanisms that can be used to guarantee and maintain a manufacturing system without defects. If a company is capable to achieve a production plan with (near) zero-defects, then an increase in the profits and customers loyalty is expected. The former is accomplished by eliminating the costs associated to scrap production and rework and the latter by providing to the customers products with the required characteristics.

At the present time, the ZDM paradigm is an open question to debate since, as usually described, attaining a zero-defect production seems an unrealistic task. However, ZDM must be analysed within distinct frames of reference according to the type of manufacturing process. In a single stage manufacturing, the ZDM paradigm must be observed at steady-state production. The methodology must be able to promote adaptability and fast recover in the presence of disturbances such as machine malfunction or degradation and differences in the incoming raw material. In multi-stage production, the attaining of ZDM is even more complex since it is necessary to track and foresee eventual small disturbances in each station that can lead, at the final production process, to a large deviation in the product quality standards. This performance improvement can be done by continuously monitor the relevant state variables scattered along the production stages and through proactive methods and computational intelligence algorithms, to reduce quality variability by anticipative reaction to possible defects. Usually, continuous improvement is made by using a repeated and cyclical questioning based on two well adopted models: PDCA (Plan-Do-Check-Act) cycle and DMAIC (Define-Measures-Analyse-Improvement-Control) Six-Sigma cycle [3]. By applying continuously these cycles, improvement becomes part of all performed activities [9].

The majority of the existing methodologies towards ZDM are concentrated on the quality of the final product instead of considering all the existing activities and/or stages within the manufacturing system. Also, to achieve ZDM in a multi-stage manufacturing system, it is mandatory to consider all the complex data relationships within a specific stage and between different stages. Having this in mind, the requirements for implementing ZDM strategies in multi-stage environments are the following:

- Distributed continuous collection of data coming from operators, equipment, in course products or final products for each stage.
- Continuous data analysis within the stage and among stages to support the real time monitoring to evaluate equipment, product and process, as well as the real-time adaptation of process and inspection parameters aiming at system improvement.
- Continuous global data analysis to support knowledge generation aiming optimisation, root cause analysis, fault diagnosis and/or fault prognosis.

The adoption of these requirements may support better decisions allowing to detect and prevent the occurrence of failures and reducing waste reaching ZDM performance and improving the productivity of a company. Multi-stage manufacturing is characterized by its distributed nature, which makes even more difficult the efficient operation of traditional rigid and monolithic control structures, since they are not, any longer, able to respond and adapt close to real time and efficiently to condition changes or process deviations. In a ZDM strategy context, the use of a MAS CPS infrastructure completely matches this distributed nature, and supports the early detection of anomalies and properly implement mitigation actions for preventing the defect generation and propagation to downstream stages and to reduce waste.

These dynamic monitoring mechanisms for the real-time and early detection of events should be complemented with global data analytics and simulation aiming to extract new knowledge and optimisation strategies supporting the analysis and control steps of the DMAIC cycle.

3 Architectural Design Principles for the Multi-agent CPS

Aiming to face the identified challenges towards ZDM strategies in multi-stage manufacturing, a distributed and intelligent infrastructure becomes mandatory to collect, monitor and process data according to the different industrial requirements, namely in terms of responsiveness and optimisation. For this purpose, the GO0DMAN CPS architecture integrates production and quality control in multi-stage manufacturing at different ISA 95 levels, by using MAS principles.

3.1 Multi-agent Systems to Implement the Distributed Intelligence

MAS play a key role to implement a distributed CPS that should be able to collect, monitor and process data in a distributed manner, at local and global levels, supporting the fast detection of process and product variability, the implementation of preventive or corrective actions, and the prediction of process faults or performance

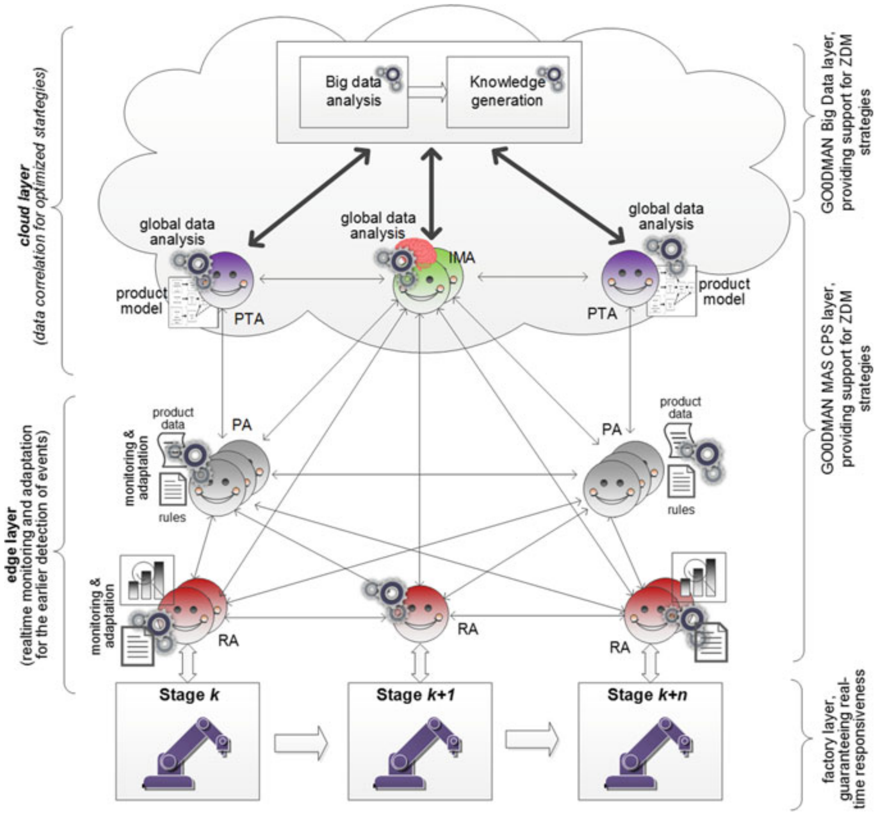


Fig. 1 GO0DMAN MAS CPS architecture for multi-stage manufacturing

degradation, allowing to design optimisation strategies for the continuous system improvement. MAS will also contribute to reach a more modular, scalable, flexible and robust system, simplifying and executing on the fly plugability and reconfigurability.

In the proposed work, the MAS CPS is based on the GRACE MAS architecture [5], and comprises a society of distributed, autonomous and cooperative intelligent agents representing the production components disposed along the multi-stage manufacturing system (see Fig. 1). In some cases, the agents have a direct association to a physical resource, e.g., a welding robot or a quality control station, but in other cases they can represent logical entities, such as data analytics tools.

These software agents interacting with physical processes are able to collect data in distributed manner. Even the products being manufactured along the line have associated an intelligent agent, which transforms a passive product into an intelligent one [4], with capabilities to gather and store data from its historical production process and current operation, supporting the real-time monitoring and decision-making.

The agents are enriched with data analytics enabling the rapid reaction to critical situations, and particularly the earlier identification of anomalies, deviations and patterns leading to condition changes or defects, and the application of proper mitigation strategies for self-adaptation and optimisation according to several criteria (e.g., adjustment of the operations' parameters). Agents placed at the higher decision level can interact with big data analytic and knowledge generation tools to get insight information related to the system optimisation based on the overall data correlation.

These agents are able to interact among each other according to some coordination patterns aiming to share their knowledge and capabilities, and consequently capable to implement inter-stage coordination policies towards the ZDM strategies. Particularly, the implementation of feedback control loops for preventing the occurrence and/or propagation of defects is achieved by the interaction among the agents.

3.2 Agents: Roles and Functionalities

The designed MAS CPS distributes the intelligence by four types of agents, identified according to the particularities of the production system: Product Type Agents (PTA), Product Agents (PA), Resource Agents (RA) and Independent Meta Agents (IMA). The existing relationships among these agents, and also their main functions, are illustrated in the UML Class diagram represented in Fig. 2.

PTAs represent the catalogue of products that can be produced in the production line and contains the process and product knowledge required to execute the

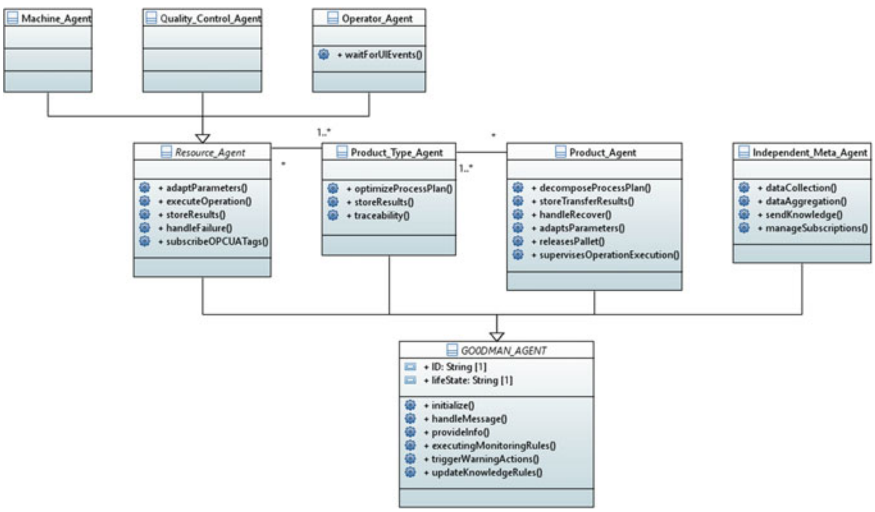


Fig. 2 UML class diagram for the GO0DMAN agent types

product, namely the process plan. These agents, 1 agent for each product model, are responsible for the following main functions:

- Launching PAs according to the production orders.
- Collection and storage of data related to the execution of the products.
- Monitoring the on-going production of products related to a specific model.
- Optimisation and adjustment of the process plan associated to the product model.
- Traceability of the executed products in the production line.

PAs manage the production of product instances in the production line (e.g., an oven or a car) according to a process plan that specifies how to execute the product. They (1 agent for each product being produced, and in case of batch production, 1 agent for each batch) are responsible for the following main functions:

- Collection and storage of production data related to the execution of the product.
- Adaptation of the process and inspection parameters of operations to be executed by the resources, according to the local knowledge and historical data.
- Monitoring the evolution of the product execution along the line to detect, amongst others, possible deviations from the plan and quality degradation.
- Pre-processing of the collected data to filter those that are sent to IMAs (mitigating the latency and bandwidth).

RAs are associated to physical resources disposed along the production line, such as robots, inspection stations and operators, and are responsible to manage the execution of their operations. Several specializations of this agent class can be considered, e.g., Machine Agents (MA), Quality Control Agents (QCA) and Operator Agents (OA). These agents, 1 for each resource, are responsible, among others, for the:

- Collection and storage of data related to the process stations, using proper IoT technologies, e.g., OPC-UA (Open Platform Communications—Unified Architecture). In case of OAs, the integration uses a proper HMI (Human-Machine Interface).
- Adaptation of the operations' parameters, according to the local knowledge and historical data from the resource perspective.
- Monitoring the evolution of the resource performance to detect, amongst others, possible quality of service degradation.
- Pre-processing of the collected data to filter those that are sent to IMAs (mitigating the latency and bandwidth problems).

The PA and RA agents uses knowledge, e.g. based on a set of rules, to monitor the process evolution and adapt the operation parameters, which is periodically adjusted by IMAs that use the results from data analysis and knowledge generation running at the cloud level. In fact, IMAs act at a strategic level to provide global system optimisation by managing the data aggregation and generation of new knowledge, as follows:

- Aggregation of the data collected in a distributed way from the system's agents.
- Management of the subscription of optimisation services requested by local agents.

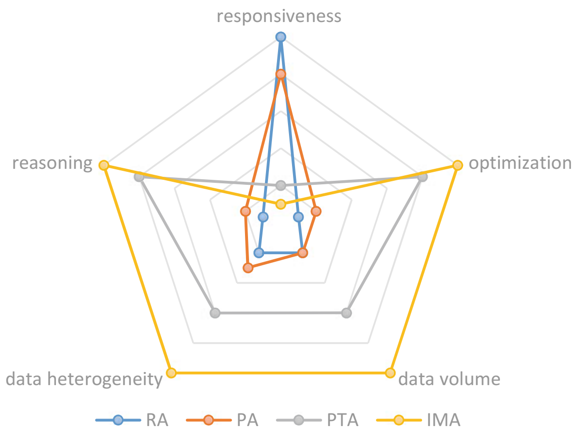


Fig. 3 Classification of the different types of GO0DMAN agents

- Monitoring the system performance to detect improvement opportunities.
- Triggering the batch data analysis aiming to identify meaningful correlations and new knowledge towards the optimisation of the production system.
- Propagation of new or adjusted knowledge to the local agents.

Since IMA agents are positioned at a strategic level and have larger response time, these agents work in a strong collaboration with big data analysis and knowledge generation tools that will run powerful data analytics algorithms considering larger datasets to achieve better optimisation and data correlation.

These agents can be classified according to several criteria regarding the decision-making, namely the responsiveness, optimisation, data volume, data heterogeneity and reasoning, as illustrated in Fig. 3.

As clearly showed, RAs and PAs focus on responsiveness by using simple reasoning algorithms to handle low data volume, while PTAs and IMAs focus on optimisation by using powerful data analytics algorithms to handle huge amounts of data without critical time restrictions.

4 Data Analysis at Edge and Cloud Levels

Data analysis algorithms, knowledge management approaches and learning techniques are applied in the proposed MAS CPS to enhance the implementation of the ZDM strategies in multi-stage manufacturing environments. Taking advantage of the MAS capabilities, the data analytics is performed in a distributed manner and spread at different levels, namely at cloud and edge levels, according to the expected temporal response and degree of optimisation, as illustrated in Fig. 4.

Local agents, i.e. RAs and PAs, operate at edge and run faster response data analysis algorithms for the knowledge extraction to allow the automatic real-time

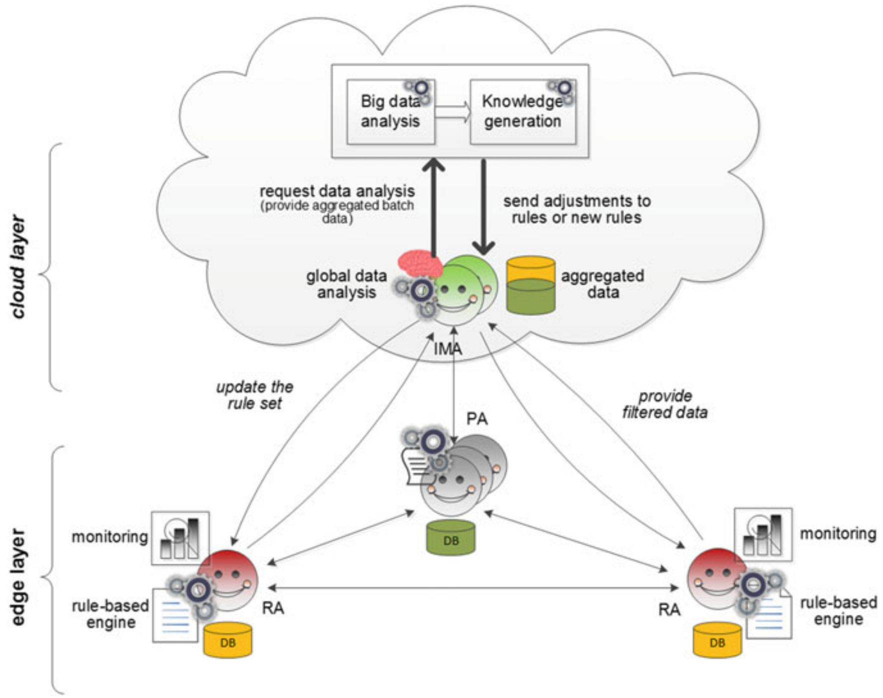


Fig. 4 Adjustment/generation of rules at cloud level and execution of rules at edge level

monitoring, detecting discrepancies between desired and actual process values, and the fast self-diagnosis, self-adaptation and predictive maintenance. For example, these agents are running a set of rules, previously defined by the cloud level, included in their knowledge base for the early detection of deviations and trends.

These rules can be atomic or composed. The atomic rules consider the variability of only one parameter, e.g., based on the well-known decision rules defined by [2] and posteriorly extended by [6] to enhance the sensitivity of control charts allowing the detection of process changes, for example:

Rule 1: IF “One point falls outside the three-sigma control limits” THEN “trigger action Type 1”
Rule 2: IF “Two out of three consecutive data points fall beyond the two-sigma warning limits” THEN “trigger action Type 2”
Rule 3: IF “Six points in a row steadily increasing or decreasing” THEN “trigger action Type 3”

Normally, the identification of the product/process variability occurs if any one or more rules from the rules-set are met. The composed rules consider the variability of more than one parameter, the dependency of which is identified by the big data analysis tool, and posterior knowledge generated by the knowledge management running in the cloud layer. An example of such kind of rule is the following:

Rule 4: IF "One point from parameter A falls outside the three-sigma control limits" AND "Four out of five consecutive points from parameter B fall beyond the one-sigma limit" THEN "trigger action Type 4"

The different parameters considered in the composed rules can come from the same stage or from different stages. In this last case, the RA needs to interact with other RAs to get information that will be used in the rule execution. However, PAs can handle the execution of composed rules comprising data from different stages in an easy manner since they have collected data related to the execution of their products (and e.g., can use it to adapt the set of functional tests to be executed at the end of the production line).

These agents are also able to supply the on-line gathered information to meta agents placed at the higher level, i.e. PTAs and IMAs, which aggregates and performs data fusion. The latest, operating at the cloud level, are running more powerful data analytics algorithms to get richer conclusions from aggregated data and allowing to generate new knowledge, e.g., adjusting the existing rules or creating new ones, that is used to optimise the predictive task of early detecting defects and preventing their propagation to downstream stages. The agents at this level provides self-optimisation and continuous improvement using also cloud based infrastructures for data storage and processing. Note that at this stage, the response time is not critical, being possible to run algorithms to deeply mine the data to find useful knowledge and correlations.

This distribution of data collection and analysis among different layers is strongly advised in the emergent industrial cyber-physical systems, where we "*should not send all collected data to be processed in the cloud but instead to make analysis in the edge*" as sustained by James Truchard from National Instruments at IFAC IMS'16, and "define data collection requirements to minimize the collection of big data" and "*enable the feedback of intelligence through the system to update control for optimal production*", sustained by Brian Weiss from NIST in the same event. Thus, the establishment of local and global levels for the data collection and analysis is the core architectural axiom when implementing ZDM strategies for the multi-stage manufacturing environment, since it allows combining the earlier detection of anomalies or process/product variability locally at each stage, with the possibility to aggregate and correlate data from different individual stages aiming to identify anomalies that can only be detected globally. The challenge is the proper balancing of the local data analysis—providing automatic and real-time monitoring and early detection of deviations and trends—and the global data analysis—providing self-optimisation and continuous improvement.

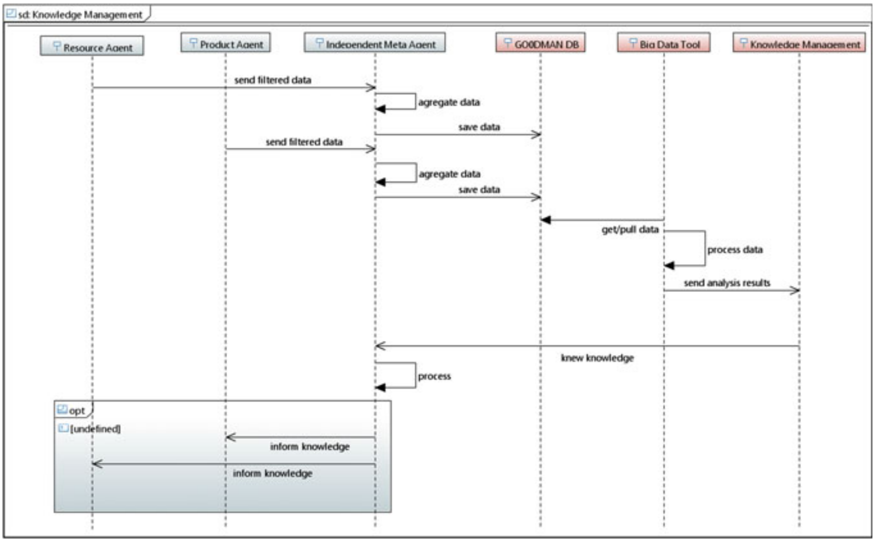


Fig. 5 Interaction among agents for the adaptation of rules based on the knowledge generation

5 Interaction Patterns Towards ZDM Strategies

The global behaviour of the multi-agent system emerges from the interaction among the distributed agents, following proper interaction patterns to achieve ZDM strategies. This section illustrates these patterns by describing the interaction pattern to support the adaptation of the rules running locally based on the generation of new knowledge and the interaction pattern to support the early detection of events.

Figure 5 depicts the message sequence diagram that supports the dynamic adaptation of rules that is based on the two-fold trigger process. First, the agents that run at the edge nodes, i.e. RAs and PAs, are continuously sending data to IMAs. All the agents are proceeding with respective ETL (Extract, Transform and Load) techniques allowing to exchange data in a efficient manner. The store of data at the IMA level (i.e. the load operation from ETL) allows for the big data tool to obtain operative data for long-term and optimized processing. After processing the data, and in the case where new, possible patterns are detected, the big data tool feeds the results to the knowledge management tool which is responsible for the generation of knowledge to the data, e.g., generating or updating rules. At this stage, the knowledge management tool informs the IMA of the changes, which on turn analyses them, decomposing the receiver agents, and informing them accordingly.

Additionally, the need for a rule update could come directly from a trigger of the PAs or RAs. These agents, by continuously managing their execution, are able to detect unknown patterns and behaviours that could need further, higher level, analysis or even rules that might be tighten, improving the productive process and consequently increase their quality (by decreasing the variability possibilities). In fact,

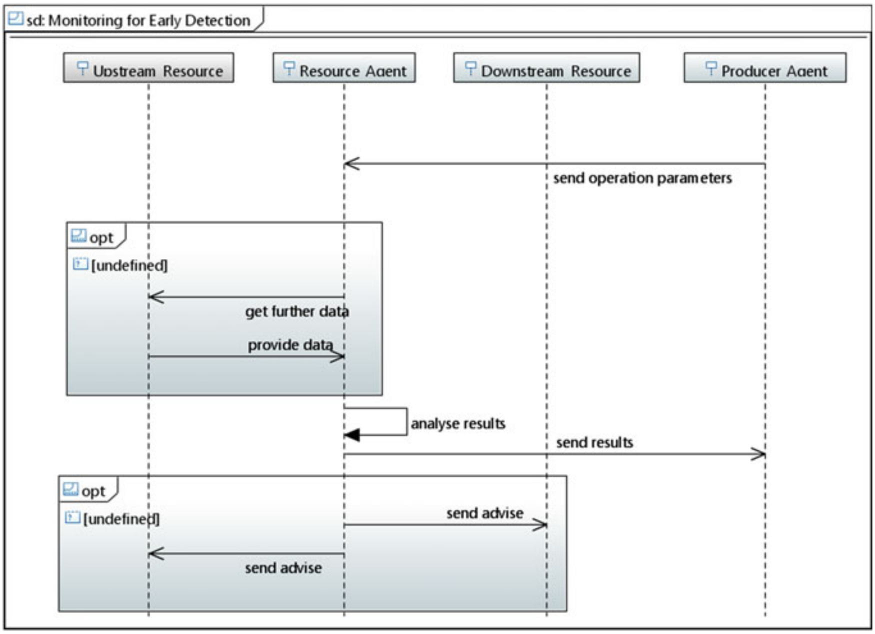


Fig. 6 Interaction among agents for the earlier detection of defects downstream propagated

when such situation appears, the designated agent requests a rule update check to the IMA, that requests such verification to the big data tool by invoking it and by providing identified data. After this step, the process is similar to the one previously described, culminating in an update or new generation of rules to the designated agent.

A second example, illustrated in Fig. 6, is related to the interaction between agents for the early detection of events considering rules that use parameters from different stages. As previously described, PAs are responsible to manage their own execution. One of its main features is the decomposition of the process plan and, for each step, the PA adapts the parameters for the next execution, considering the previous execution parameters. Therefore, the PA, before issuing a processing initialization sends the operation parameters to the RA.

After executing the operation, the RA analyses the operation results and sends them to the PA. At this stage, both the PA and the RA consider this new output into their life-cycle aiming at different perspectives. The RA considers this to properly analyse its own process execution, enabling a continuous process optimization and to detect future problematic situations at the physical resource level. Additionally, the RA cooperate with upstream and downstream RAs, advising them to proceed with corrective measures enabling the whole process quality standards to be meet, although its execution quality is being degraded. On the other side, the continuous monitoring of the PA enables to forecast and predict future product quality issues, allowing a parameter adjustment for the subsequent RAs, mitigating problematic situations, while maintaining and increasing the product quality. As an example, PAs can simply adjust the parameters of the next operation or the set of texts to be executed in the inspection station.

6 Conclusion

Quality management can be considered a key concern of almost manufacturing organizations since high-quality products can give an organization a significant competitive advantage. Good quality products reduce the costs of rework, waste, complains and returns and, most important, generates satisfied customers. The majority of the existing quality control and improvement methodologies for multi-stage manufacturing systems were built upon quantitative modelling of the system (see [8] for more details). Due to its inherent complexity, this type of manufacturing system presents significant challenges and opportunities, at both operational and tactical levels, aiming to achieve a (near) zero defect manufacturing. One of the challenges is to continuously collect, monitor and analyse huge amounts of data within a single stage and among different stages to periodically generate knowledge. The generated knowledge will support better decisions allowing to detect and prevent the occurrence of failures reaching ZDM performance and improving the productivity of a company.

In this context, this paper introduces the main principles of a multi-agent CPS aiming at applying ZDM in multi-stage manufacturing systems. This concept is being developed under the EU H2020 GO0D MAN project. In this article, the MAS architecture was introduced by analysing the roles and functions of each agent, the way these agents are distributed along two layers, the roles of data analysis in each layer, and finally the way the overall system emerges from the interaction among individual agents. At this point, two examples of interaction patterns towards the implementation of the ZDM strategies were illustrated, namely the interaction among agents placed at the edge level for the early detection of defects, and the interaction between agents placed in cloud and edge levels for knowledge generation.

Future work is related to the further specification of the multi-agent CPS architecture concerning the interfaces with data analytics tools and physical process and inspection equipment and legacy systems, and the implementation of the designed solution by using a proper multi-agent development framework, e.g., the Java Agent Development Framework (JADE) platform [1].

Acknowledgements

This work is part of the GO0D MAN project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 723764.

References

1. Bellifemine, F., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. Wiley (2007)
2. Electric, W.: Statistical Quality Control Handbook. Western Electric Corporation (1956)
3. Gryna, F., Chua, R., DeFeo, J.: Juran's Quality Planning and Analysis, 5th edn. McGraw-Hill (2006)
4. Leitão, P., Rodrigues, N., Barbosa, J., Turrin, C., Pagani, A.: Intelligent products: the GRACE experience. Control Eng. Pract. **42**, 95–105 (2015)
5. Leitão, P., Rodrigues, N., Turrin, C., Pagani, A.: Multi-agent system integrating process and quality control in a factory producing laundry washing machines. IEEE Trans. Ind. Inf. **11**(4), 879–886 (2015)
6. Montgomery, D.C.: Introduction to Statistical Quality Control, 6th edn. Wiley (2009)
7. Shi, J.: Stream of Variation Modeling and Analysis for Multistage Manufacturing Processes. CRC Press (2007)
8. Shi, J., Zhou, S.: Quality control and improvement for multistage systems: a survey. IEEE Trans. 744–753 (2009)
9. Slack, N., Brandon-Jones, A., Johnston, R.: Operations Management, 7th edn. Pearson (2013)
10. Wang, K.S.: Towards zero-defect manufacturing (zdm)—a data mining approach. Adv. Manuf. **1**(1), 62–74 (2013)
11. Wooldridge, M.: An Introduction to Multi-Agent Systems. Wiley (2002)

Multicriteria Supplier Selection for Supply Chain Management

Fouzia Ounnar, Olivier Labarthe, Patrick Pujo and Benoit Montreuil

Abstract Under the pressure of global competition, cycle time reduction and increasing product complexity, enterprises have realized that internal improvements are certainly important but not sufficient. Company performances depend not only on internal activities but also on external resources from supply networks made up of customers, suppliers, and subcontractors. In the present paper, the conventional choice of suppliers involved in a supply network is reconsidered. The process proposed to support decisions in a context of choice is based on the exploitation of a Simulation of Extended Enterprises 'SEE' platform for the configuration of a multicriteria method (Analytic Hierarchy Process 'AHP') and the analysis of the choice of suppliers. This multicriteria method, restructured into two phases, is presented and the implementation process is described. The case study focuses on snowmobile assembly with 75 workstations in a multi-product assembly line, 420 supplied parts and 48 suppliers. Multiple simulation cases are presented and the obtained results are analysed.

Keywords Supply chain management • Supplier evaluation • Analytic hierarchy process • Multicriteria decision making • Logistics network simulator

---

F. Ounnar (✉) · P. Pujo
Aix-Marseille Univ., CRET-LOG, Aix-en-Provence, France
e-mail: fouzia.ounnar@univ-amu.fr

P. Pujo
e-mail: patrick.pujo@univ-amu.fr

O. Labarthe
University of Bordeaux, IMS, Bordeaux, France
e-mail: olivier.labarthe@u-bordeaux.fr

B. Montreuil
ISyE, Georgia Institute of Technology, Atlanta, USA
e-mail: benoit.montreuil@isye.gatech.edu

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_3

1 Introduction

In the overall context of procurement, production and distribution, the performance of the enterprises depends not only on the improvement of their internal activities but also on the use of external resources. In response to increasing global competitive pressure, supply networks made up of customers, suppliers, and subcontractors have emerged.

Supply chains are generally defined on the basis of contractual relationships, each customer defining a list of suppliers and each supplier identifying their potential customers [1]. Customer/supplier relationships are defined according to a bilateral agreement with reciprocal commitments on volumes and/or periods and/or deadlines. Thus, the suppliers have a partial or total knowledge about future orders, based either on forecasted orders or on contractual commitments.

However, in such context, divergent interests may occur when customers are constrained by their suppliers and vice versa. The dynamics of a supply chain seems very weak. Therefore, the deterioration of the relationship between two links in the supply chain following a disturbance may cause substantial dysfunctions upstream and downstream. Companies are thus evolving towards new network-oriented organizational forms; [2–5] studied various problems linked to the design and management of flows in logistic networks.

One of these problems is the evaluation of suppliers in order to select the best one. In this type of problem, the methods exploited are mainly of mono-criterion type often focussing on total investment, procurement or operating costs, neglecting the costs of supplier invoicing, delivery delays and delivered product quality. This observation comes from analysing various literature reviews that deal with supply chain simulations [6–8].

The past years have witnessed a strong evolution of the relationships between companies so that each individual partner achieves a better internal management and all partners contribute to a better performance in meeting the customers' expectations [9].

In this paper, the conventional choice of suppliers involved in a supply network [10, 11] is reconsidered. The main assumption is that a forecasted organization defined on the basis of contractual commitments can change over time. Thus, the choice of a supplier for a given component will be made according to the actual state of his performances (such as: production system state, delivery times, and supply disruptions) [2].

The Analytic Hierarchy Process (AHP) method [12] is considered to take into account both qualitative and quantitative factors to facilitate supplier evaluation and selection. This multicriteria method is deployed on the platform for the Simulation of Extended Enterprises (SEE) developed by [13] at the CIRRELT Research Centre. The SEE platform facilitates the definition, implementation and simulation of various supplier behaviours to conduct evaluations and selections [14]. The choice of the SEE platform and the AHP method comes from meeting the skills of the two research teams presenting this work. The goal is to validate the feasibility of such approach and highlight the different steps to be implemented for the multi-criteria supplier evaluation based on supply chain simulation.

This proposal is in line with research topics such as Physical Internet [15] and intelligent control solutions [16].

First of all, the presentation of the AHP method is performed in two phases and the general architecture of the simulation platform is introduced. In order to present the use of AHP for supplier evaluation in the industrial case study, the definitions of the supplier behaviours are presented and the AHP model as well as its implementation is performed. Then, the various simulation results are described and analysed. Finally, concluding remarks and research prospects are provided.

2 Exploitation of the Decision-Making Method in a Simulation Environment

In this section, the presentation of the AHP method is performed into two phases and the SEE platform used for the different experiments is presented.

2.1 The AHP Method: Configuration and Exploitation Phases

The AHP method can be implemented into two phases: a (static) configuration phase followed by a (dynamic) exploitation phase [17]. This is necessary in the case of distributed, automated and real-time use of AHP, as we have proposed for operation control. In operation management, the decisions are usually made with foresight, in the form of planning and scheduling. These decisions are usually taken regarding a mono-criterion optimization that avoids dealing a part of reality [18, 19]. They are extremely sensitive and can be quickly challenged for various reasons: no data updates, appearance of disturbances, changing priorities… It is therefore more interesting to consider the different criteria interacting in decision-making, and this in real time. However, AHP requires interaction with experts (decision-makers) to establish the hierarchical structure of decision-making and validate the consistency before seeking the best possible compromise. This description of the decision problem can be carried out once, until validation of the decision-making structure, and then be reused at each search for a compromise coming from an evolution of the data set related to the alternatives.

This structuring of AHP into two phases allows automatic and real-time exploitation of multicriteria decision-making. This latter was implemented in the context of a holonic and isoarchic control [20]. The use of the multicriteria decision method (AHP) into two phases then finds its interest. Indeed, the resulting compromise from the final classification depends on the setting of the parameterization of the relative importance of the criteria and their indicators, and therefore of the weights allowing to perform the pairwise comparison of alternatives. Indeed, before using the AHP algorithm to rank the alternatives, it is necessary to validate the configuration. Parameter setting allows the relative importance of the criteria and their indicators to be adjusted: this is the static phase of the algorithm. Once the setting is done and validated, the dynamic exploitation of the AHP method will allow ranking automatically the alternatives. This has been implemented on a HMES (Holonic Manufacturing Execution System) [21]. A similar principle is applied here to a multi-agent system—the SEE platform.

2.2 *Presentation of the SEE Platform*

In order to understand the decisions and the interactions within value-added networks, simulation is naturally used by the decision-makers. The Simulation of Extended Enterprises (SEE) platform facilitates the definition and the implementation of the behaviours of various suppliers in view of their evaluation. The case study focuses on the assembly of snowmobile with 75 workstations along a multi-product assembly line, 420 supplied parts and 48 suppliers.

SEE is a holistic modelling and simulation platform considering all the actors and flows within manufacturing and logistics networks. Modelling and simulation rely on the agent paradigm for the representation of the actors' behaviour [13, 22]. Two main modules make up the architecture of the SEE: the Simulation module (Fig. 1, left-hand side) and the Viewers module (Fig. 1, right-hand side). In each simulation, a set of autonomous agents interact with each other. The main actors of the networks that are modelled by the agents are the following: the customers, the retailers, the production planners, the purchasers, the production managers, the distributors of finished products to the retailers, the transportation planners, the transportation dispatchers and the suppliers. The interaction links that are established between the agents are represented in Fig. 1.

The Viewers are dedicated to the display during simulation of data related to the different aspects of the logistics and manufacturing network. The interfaces developed include a geographic interface, a statistical interface, a financial interface, an assembly line interface, a time-dependent location interface, a supply chain interface, a transport interface, and a supplier interface [13, 23].

Within the framework of this study, the multicriteria evaluation concerns the performances of the suppliers. The supplier agents interact with the purchasing agent and the production planner agent. The purchasing agent receives the production plans and confirms, from the list of suppliers, whether or not the parts needed to build the products will be available on time. The agent informs the production planner agent whether this plan is infeasible according to the supply constraints. The production planner agent can then modify the initial plan and submit the modified plan until a feasible plan is obtained. According to a feasible

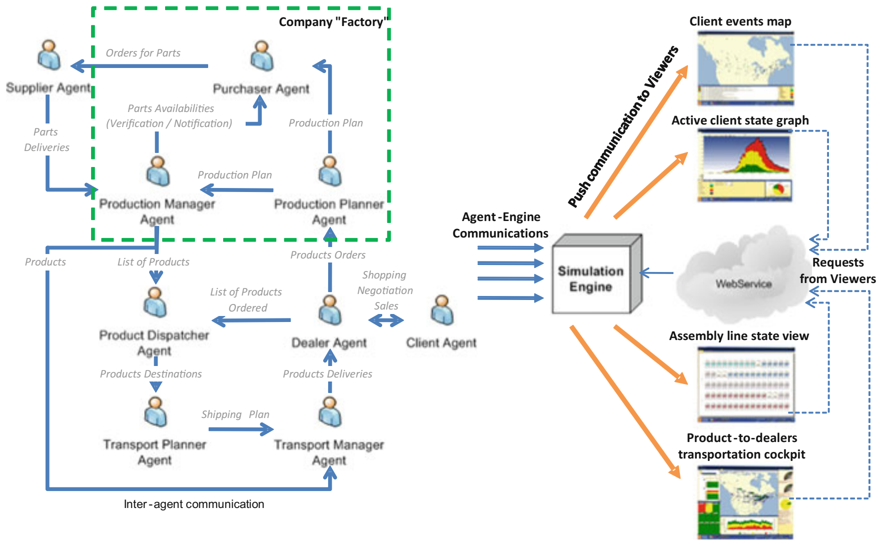


Fig. 1 General architecture of the SEE platform

production plan and depending on the state of the inventory (information sent by the production planner agent) the purchasing agent orders the parts from the different suppliers. The production planner agent simulates the assembly line, respecting the production plan. Before launching a product on the assembly line, the production planner agent verifies that all the necessary parts are in inventory.

In this platform, the choice of supplier is done statically with one supplier per part. The objectives of the implementation and use of the AHP method are to determine, dynamically and using several criteria, the best supplier among a list of potential suppliers for each part purchased.

3 Implementation of the AHP Method in the SEE Platform

This section is dedicated to the presentation of the process of implementation of the AHP method within the SEE platform. Implementing this multicriteria decision method in an agent-based supply chain simulation requires the definition and the specification of adapted behaviours and interactions. A first sub-section is dedicated to the description of supplier behaviours. The supplier agent profiles as well as their interactions with agents simulating the decision centres of the other organizational units are defined. Then the construction of AHP model is detailed in the context of this study.

3.1 Profiles and Behaviours of the Supplier Agents

The supplier agents receive orders from the purchasing agent and must deliver the requested parts to the manufacturing company. The supplier agents are subject to several lead times: order processing lead-time, product availability lead time, shipping lead time and transportation lead time. The sum of these lead times constitutes the supplier's response time, i.e., the time elapsed between the emission of the order and the reception of the ordered products.

Each supplier agent has a brief description including the name of the supplier he represents, the supplier's geographical position as well as a profile defining the supplier behaviour. A profile contains the definition of multiple elements characterizing the behaviour of a supplier considered during the simulation: (i) a pattern of order processing lead-time, (ii) a pattern of delivery lead-time, (iii) a percentage of splitting orders, (iv) a percentage on delivered part quality, and (v) the supplier's activity level.

The patterns of order processing lead time define the limits in terms of lead time to purchase or to produce the desired goods according to different processing modes (fast and reliable, fast and not reliable, slow and reliable, slow and not reliable or almost fixed date). From this pattern, probabilities of order processing lead-time according to a Weibull distribution will be associated to each part or product that can be ordered from the supplier. Similarly, the supplier's delivery lead time defines the limits in terms of lead time to deliver the goods ordered according to different delivery modes (fast and reliable, fast and not reliable, slow and reliable, slow and not reliable or in almost fixed lead time). From this pattern, probabilities of delivery lead-time according to a Weibull distribution will be associated to each supplier. The combination of these two patterns defines the time necessary for a supplier to fulfil an order. The quality of the delivered products, defined as a percentage of the quantity ordered, corresponds to the non-functional or damaged products received by the manufacturing company. Each supplier is also defined by a level of activity or collaborative (passive or proactive) behaviour. A passive supplier does not send information to his customers. A proactive supplier informs his customers if he expects to be unable to meet the delivery deadline. Suppliers are listed based on the production and delivery mode they use: make-to-stock/deliver-to-order, or make (and deliver) to order.

In the first level of simulation, there is only one supplier for each part or product that could be ordered. In each supply scenario, the set of suppliers is generated randomly according to the parameters that define each supplier's profile.

3.2 Implementation of the AHP Method

The objective of the implementation of the AHP method is to select the suitable supplier from a list of potential suppliers, dynamically and for each part.

The implementation can then proceed through the two main phases (Configuration and Exploitation phases, presented in Sect. 3.1).

3.2.1 Phase 1: Configuration (Static Phase)

Step 1*: *Definition of the hierarchical structure

The AHP hierarchical decision structure is established from the information related to the SEE simulation platform of manufacturing networks (from manufacturer to retailers) and its sales interactions with the customers (CRSNG-Bell-Cisco, Research Chair in eBusiness Design project). Information is retrieved from [13]. The description of the case study considered (75 workstations in a multi-product assembly line, 420 supplied parts, 48 suppliers) allows the following decision structure (Fig. 2) to be built.

Step 2*: *Pairwise comparison of the elements of the structure

Once the hierarchical structure is elaborated, the matrices of pairwise comparison of the various elements (criteria and indicators) are defined. Indeed, during the AHP configuration phase, the expert builds matrices indicating his preferences, using the Saaty scale (1, 9) [12]. Having three criteria, three indicators for the second criterion and five indicators for the third criterion, three pairwise comparison matrices are calculated (Table 1). This setting requires assessment by an expert, who can be specific to each supplier.

Step 3*: *Evaluation of the relative importance between the various elements

During this step, the relative importance of the various elements (criteria and indicators) is calculated. The results are presented in Table 2.

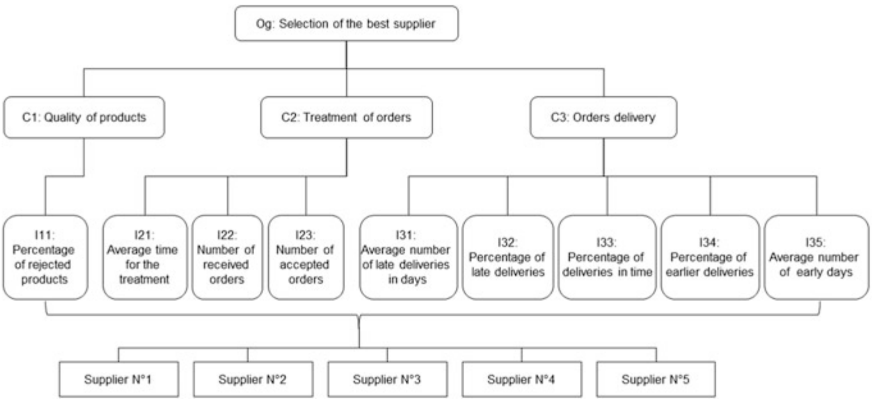


Fig. 2 Hierarchical structure of the supplier selection

Table 1 Matrices of pairwise comparisons

	Pair wise comparison matrix for the three Criteria (CC)		
	C1	C2	C3
C1	1.000	0.333	0.143
C2	3.000	1.000	0.200
C2	7.000	5.000	1.000

Table 2 Relative weights of the various elements

CC matrix	
	CrOg
C1	0.07902426
C2	0.22488756
C3	0.69608053

3.2.2 Phase 2: Exploitation (Dynamic Phase)

Step 4*: *Pairwise comparison of the elements of the structure

(Level 4 compared to Level 3 in the hierarchy)

The values used at this stage correspond to the real values of the indicators of the alternatives. The values are obtained in real time during the simulation.

Step 5*: *Evaluation of the relative weights between the various elements

(Level 4 compared to Level 3 in the hierarchy)

The procedure is identical to the one used during the configuration phase.

Step 6*: *Aggregation of the relative weights to establish a ranking of alternatives

The aggregation principle consists in carrying out matrix products. The result is a vector providing the relative importance of the alternatives with regard to the global objective. This vector associates a value to each of the suppliers. The supplier with the highest value is the one who best satisfies all the criteria.

4 Impact of the Integration of AHP as a Multicriteria Decision Aid Method

This section presents and compares the results obtained by simulation according to two distinct procurement configurations. Its main purpose is the study of the purchasing agent behaviour, this study being based on simulation cases for the process of decision-making for the selection of a supplier for a given part. For the first procurement configuration, each component is assigned to one of the 48 selected suppliers (Simulation Case 0). Then, for the second procurement configuration, five

Table 3 Parameters of the weights assigned to the different cases

	C1	C2	C3	I11	I21	I22	I23	I31	I32	I33	I34	I35
Case 1	0.079	0.225	0.696	1	0.481	0.082	0.232	0.031	0.088	0.478	0.251	0.152
Case 2	0.010	0.980	0.010	1	0.481	0.082	0.232	0.031	0.088	0.478	0.251	0.152
Case 3	0.010	0.010	0.980	1	0.481	0.082	0.232	0.031	0.088	0.478	0.251	0.152
Case 4	0.079	0.225	0.696	1	0.010	0.980	0.010	0.031	0.088	0.478	0.251	0.152
Case 5	0.079	0.225	0.696	1	0.481	0.082	0.232	0.010	0.010	0.960	0.010	0.010

potential suppliers among the set of 48 suppliers are defined for every part (Simulation cases 1, 2, …, 5 (Case 1–5), see Table 3). The use of the AHP method relies on the identification of the best supplier among the five potential suppliers.

In order to compare and assess the results obtained, the second sub-section is dedicated to the definition of performance indicators and the presentation of significant results.

4.1 Description of the Simulation Cases

The first supply configuration is the simplest decision-making process; since there is no possible choice for the purchasing agent to make (a part is associated with one supplier only). The second supply configuration is dedicated to the application of AHP method, considering the values of criteria and indicators defined in Sect. 4 and presented in Table 3. Five simulations cases are considered:

- **Case 1**: The set of criteria is used (AHP case),
- **Case 2**: Criterion C2 (Order processing) is preferred (AHP case),
- **Case 3**: Criterion C3 (Order delivery) is preferred (AHP case),
- **Case 4**: Indicator I22 (Number of orders received) is preferred (AHP case),
- **Case 5**: Indicator I33 (Percentage of on-time deliveries) is preferred (AHP case).

Various simulation cases are executed in the procurement configuration using the AHP method, the relative importance of various criteria and indicators being modified.

4.2 Results Analysis

The simulations and the associated results facilitate the comparisons of the supplier agents' behaviours and make it possible to observe the impact of the supplier selection process on the assembly line performances. In order to compare the two procurement configurations—one supplier per part versus five potential suppliers per part and applying a multicriteria decision method—three significant indicators are proposed to compare the performances of the different simulations.

- **Number of orders received per supplier**: The number of orders received per supplier is quite different from one case to another. In the first case, a homogeneous distribution of orders per supplier is observed. In the other cases, the AHP method allows the “best” supplier to be dynamically selected in terms of several criteria and indicators. The distribution of orders decreases according to the weight associated to a criterion or an indicator. The selection of suppliers and the distribution of orders per supplier are directly correlated with the actual performances. Nevertheless, there is a difference when criterion C3 (order delivery) is privileged, i.e., the number of orders received is above average for the suppliers at “the bottom of the list” and exceeds one hundred for supplier 38. Indeed, these suppliers obtain a lower score for criterion C2 (order processing), and thus focusing on order delivery gives them an advantage.
- **Average rank of a supplier**: This average, calculated each time a new order comes from the purchaser agent, considers all the ranks provided by AHP for each supplier (48 suppliers). For example, a supplier having delivered three batches of different parts at a given date and having been ranked by AHP 1st, 3rd and 5th, respectively, has an average rating of 3 at that date. A supplier with an average rank closer to 1 is considered as “highly solicited”. An average rank close to 5 indicates a supplier often ranked last with the AHP algorithm. The list established by the purchaser agent based on the second procurement configuration shows the fluctuation of supplier ranks when several criteria are considered (Fig. 3). Indeed, depending on the state of his production system, the supplier will be more or less efficient. The performance of each supplier is highlighted through the multicriteria method, which replaces the experience and the behaviour of the purchaser. For six delivery dates, four suppliers with significant behaviours are represented, showing that the initialization phase will have an impact until the third delivery; then the real performances modify the ranking.

From the indicator ‘average rank of a supplier’, it is possible to show the impact of the weights of the various criteria and indicators in the process of alternatives evaluation. The analysis of the curves related to the average rank of each supplier

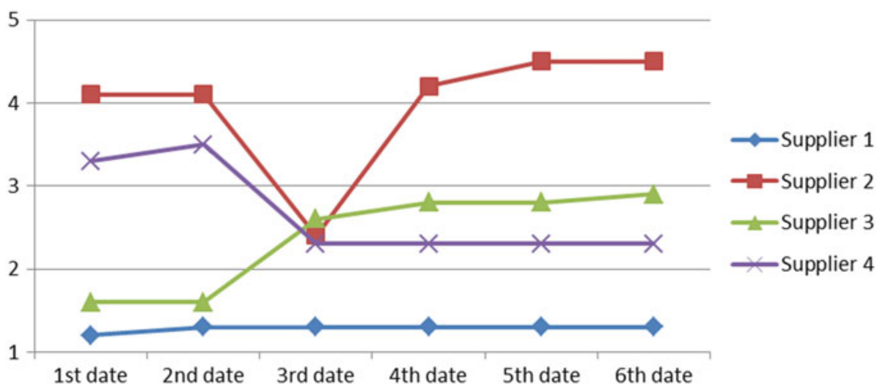


Fig. 3 Evolution of the average rank of four suppliers with the AHP method

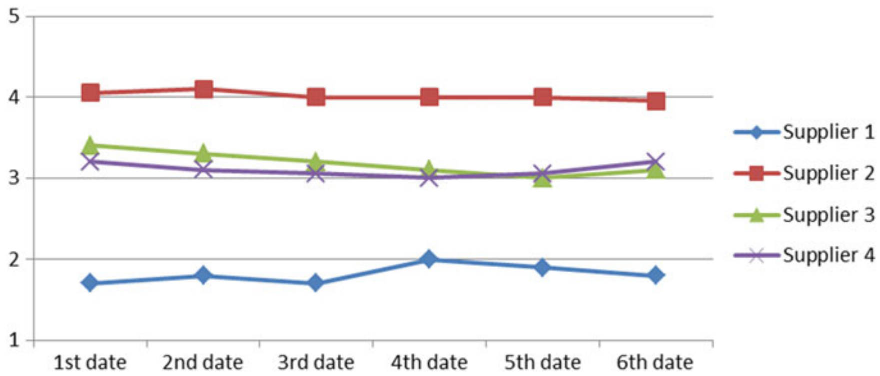


Fig. 4 Evolution of the average rank of four suppliers (Case 2)

(alternative) in the second procurement configuration, where the relative importance of criteria and indicators is different (Case 1, 2, 3, 4, and 5), shows that adjusting the importance of the criteria and indicators influences the final behaviour of the supply chain. In Case 2, the ranks of the suppliers are only slightly modified, which demonstrates that order processing evolves little over time (Fig. 4). Supplier ranks in Case 3 evolve markedly since the list is re-evaluated after each order reception. Case 4, where the indicator of the number of received orders is privileged, shows a similarity with Case 1. For Case 5 (where indicator I33, related to the percentage of on-time deliveries, is preferred), the average rank is quite stable compared to that in Case 1, given that the suppliers late in delivering are rarely chosen during these simulations.

- **Delays on the assembly line**: The indicator 'numbers of delays' obtained for the simulations of the six cases are compared in Fig. 5.

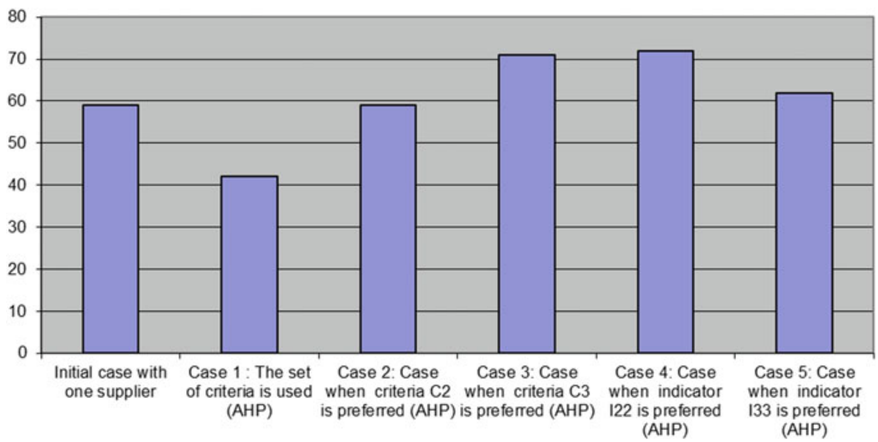


Fig. 5 Number of delays on the assembly line

If all the delays occurring over a period of one month are added together, the simulation using the AHP method and considering all of the criteria (Case 1) generates a shorter combined delay than the other cases (29% improvement over the initial case).

5 Conclusion and Future Works

The paper proposes the application of the AHP method for supporting the decision-making process related to the choice of the best supplier among a set of potential suppliers in the context of a multi-agent simulation of enterprises. The exploitation of the AHP method into two phases and in an automatic and real-time manner is discussed. In this approach, the supply chains are built gradually as the choice of suppliers is completed. The dynamics induced is due to the flow organization resulting from the coordination and cooperation relations existing temporarily between partners: what is optimal at a given time is not necessarily optimal later and should be reassessed.

The first results highlight the benefits of exploiting a multicriteria decision aid method for the selection of suppliers. Moreover, the approach proposed allows the decision maker to test various possible settings for the AHP method and to define the parameters to be used within the framework of an industrial partnership; these parameters can of course be updated.

These first results demonstrate that the simulations results are highly improved by the addition of the AHP method for supplier evaluation to the SEE platform. Assembly line delays caused by late deliveries are reduced and the purchasing agent's behaviour is now much more realistic. The simulations using the AHP method achieve better results in terms of the number of delays recorded, the number of orders received and the rank of each supplier.

These results were expected, since they confirm the AHP contributions to obtain effective workshop control solutions [18].

The contribution proposed in this work allows the SEE platform to take into account the supplier data in the decision process. It should be noted that other criteria—such as the level of customer satisfaction, the costs generated and the delivery lead times for the finished products—could also be considered in this context. These new criteria could be taken into account in the decision process. This can be done by adding the criteria and their associated indicators to the structure described in the present work. This would allow the purchasing agent to take into account not only the data received from the suppliers but also the data sent by the customer agents downstream of the supply chain.

References

1. Slack, N., Chambers, S., Johnston, R.: Operations and process management: principles and practice for strategic impact, Pearson Education (2009)
2. Ounnar, F., Pujo, P.: Supplier evaluation process within a self-organized logistical network. Int. J. Logist. Manage **16**(1), 159–172 (2005)
3. Leitão, P.: Agent-based distributed manufacturing control: A state-of-the-art survey. Eng. Appl. Artif. Intell. **22**(7), 979–991 (2009)
4. Bilgen, B.: Application of fuzzy mathematical programming approach to the production allocation and distribution supply chain network problem. Expert Syst. Appl. **37**, 4488–4495 (2010)
5. Montreuil, B.: Production planning optimization modeling in demand and supply chains of high-value products. In: Logistics Systems: Design and Optimization. Kluwer Academic Publishers (2004)
6. Terzi, S., Cavalieri, S.: Simulation in the supply chain context: a survey. Comput. Ind. **53**(1), 3–16 (2004)
7. Monostori, L., Vancza, J., Kumara, S.R.T.: Agent-based systems for manufacturing. CIRP Annals Manufact. Technol. **55**(2), 697–720 (2006)
8. Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L.K., Young, T.: Simulation in manufacturing and business: a review. Eur. J. Oper. Res. **203**(1), 1–13 (2010)
9. Montreuil, B., Labarthe, O., Cloutier, C.: Modeling client profiles for order promising and delivery. Simul. Model. Pract. Theory **35**, 1–25 (2013)
10. Vonderembse, M., Tracey, M., Tan, C.L., Bardi, E.J.: Current purchasing practices and JIT: some of effects on inbound logistics. Int. J. Phys. Distrib. Logist. Manag. **25**(3), 33–48 (1995)
11. Cho, N.: Understanding supply network coordination in electronic supply network coordination. In: Mahdavi, I., Mohebbi, S., Cho, N. (eds.) Intelligent and Dynamic Environments: Modeling and Implementation. IGI Global, pp. 1–23 (2011)
12. Saaty, T.L.: The Analytic Hierarchy Process. McGraw Hill, New York (1980)
13. Montreuil, B., Cloutier, C., Labarthe, O., Loubier, J.: Agent-oriented modeling of distributed supply chains. In: Proceedings of the International Conference on Information Systems, Logistics & Supply Chain (2006)
14. Montreuil, B., Cloutier, C., Labarthe, O., Loubier, J.: Holistic modelling, simulation and visualisation of demand and supply chains. Int. J. Bus. Perform. Supply Chain Model. **7**(1), 53–70 (2015)
15. Montreuil, B.: Towards a physical internet: meeting the global logistics sustainability grand challenge. Logist. Res. **3**(2–3), 71–87 (2011)
16. Cardin, O., Ounnar, F., Thomas, A., Trentesaux, D.: Future industrial systems: best practices of the intelligent manufacturing and services systems (IMS2) French research group. IEEE Trans. Ind. Inf. **13**(2), 704–713 (2017)
17. Ounnar, F., Pujo, P.: Pull control for job shop: holonic manufacturing system approach using multicriteria decision-making. J. Intell. Manufact **23**, 141–153 (2012)
18. Ounnar, F., Pujo, P., Hachicha, M., Dubromelle, Y.: Study of an intelligent and multicriteria scheduling service, using academic benchmarks. Int. J. Comput. Integr. Manuf. **29**, 1166–1176 (2015)
19. Pujo, P., Ounnar, F., Power, D., Khader, S.: Wireless holon network for job shop isoarchic control. Comput. Ind **83**, 12–27 (2016)
20. Pujo, P., Broissin, N., Ounnar, F.: PROSIS: an isoarchic structure for HMS control. Eng. Appl. Artif. Intell. **22**(7), 1034–1045 (2009)

21. Dubromelle, Y., Ounnar, F., Pujo, P.: Service oriented architecture for holonic isoarchic and multicriteria control. In: Service Orientation in Holonic and Multi-Agent Manufacturing Control, Studies in Computational Intelligence, vol. 402. Springer (2012)
22. Labarthe, O., Espinasse, B., Ferrarini, A., Montreuil, B.: Toward a methodological framework for agent-based modelling and simulation of supply chains in a mass customization context. Simul. Model. Pract. Theory **15**(2), 113–136 (2007)
23. Hakimi, D., Montreuil, B., Labarthe, O.: Supply Web Agent-Based Simulation Platform. In: Proceedings of the 3rd International Conference on Information Systems, Logistics and Supply Chain, Casablanca (2010)

Environmental Assessment Using a Lean Based Tool

Andrés Muñoz-Villamizar, Javier Santos, Jairo R. Montoya-Torres and Marta Ormazábal

Abstract The increase on environmental awareness has triggered numerous green initiatives at companies. This paper presents the Overall Greenness Performance (OGP), a metric based on the well-known Overall Equipment Effectiveness (OEE). The OGP is a hierarchy of metrics based in the Lean Manufacturing framework that measure how environmentally efficient a manufacturing operation is executed. The purpose of the metric is to quantify improvements in productive and environmental efficiencies, relative to a company's initial situation, after implementing a lean-green manufacturing system.

Keywords Environmental · Performance · Efficiency · Metric
Lean

A. Muñoz-Villamizar · J. Santos · M. Ormazábal
TECNUN Escuela de Ingenieros, Universidad de Navarra,
Manuel de Lardizábal 15, San Sebastián, Spain
e-mail: afmvillamizar@tecnun.es

J. Santos
e-mail: jsantos@tecnun.es

M. Ormazábal
e-mail: mormazabal@tecnun.es

J. R. Montoya-Torres (✉)
Facultad de Ingeniería, Universidad de La Sabana,
km 7 Autopista norte de Bogotá, D.C., Chia (Cundinamarca), Colombia
e-mail: jairo.montoya@unisabana.edu.co

A. Muñoz-Villamizar
Escuela de Ciencias Economicas y Administrativas, Universidad de La Sabana,
km 7 Autopista norte de Bogotá, D.C., Chia (Cundinamarca), Colombia

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_4

1 Introduction

Increasingly, companies are taking initiatives to mitigate the negative social and environmental impacts to address emerging needs of society. The assumption that natural resources are infinite and regenerative capacity of the environment is able to compensate for all human action is no longer acceptable [1]. Thus, organizations are becoming progressively more aware of their operations' impacts on people, the planet and profits (triple bottom line perspective), and are under increasing pressure to account for their resource consumption and environmental footprint [2].

On the other hand, manufacturing sectors are noted for their material consumption and waste generation [3]. Thus, the interest in sustainable manufacturing has encouraged a lot of research on the development of decision-making tools, metrics and sustainability measurement systems [4].

Researchers agree that companies can achieve a greater competitive advantage by making their businesses more environmentally friendly [5]. That is, green management practices may improve the efficiency and competitiveness of a company by reducing its environmental risks and impacts. However, despite the importance of green approach, many companies are still skeptical about the business benefits of green systems. The issue of how to address the environmental management practices may be a costly endeavor the economic factor is not considered [6], but above all, if it is not correctly quantified.

Thus, monitoring, measurement and analysis play a significant role in integrating the environmental and economic elements. Selecting meaningful and effective tools for measuring production and environmental efficiency is important due to the aforementioned perception that implementing environmental options and complying with regulatory and public pressures increases costs [7]. However, the company's environmental efficiency cannot be measured independently of production efficiency or isolated from the company's context [8].

This paper proposes a new metric for the assessment of the environmental performance of a manufacturing company, called Overall Greenness Performance (OGP). The paper is organized as follows. Section 2 presents a review of the related literature. Section 3 describes the metric, including some considerations of the problem under study. Section 4 closes the paper with some concluding remarks and highlights some opportunities for further research.

2 Literature Review

2.1 Sustainable Manufacturing

Sustainability can be defined as "meeting the needs of current generations without compromising the ability of future generations to meet their needs in turn" [9]. This concept, first introduced in 1987 can be witnessed in the subsequent emergence and adoption of environmental practices and standards, either in relation to production (life cycle analysis, green building standards, etc.) or to management procedures (environmental management system) in industry [8].

There is a worldwide effort to practice sustainability in businesses, including engineering, design, and manufacturing. Among the businesses with sustainable goals, manufacturing sectors generally require the most change due to their seriously negative impact on environment [3]. For instance, according to the report of IEA (2015), manufacturing industries are solely responsible for 36% of carbon dioxide emissions and they consume approximately one-third of the globe's energy consumption.

The concept of Life Cycle Analysis (LCA) has been adopted in order to assess and improve the environmental performance of goods and services [10]. Many different methodologies, based in LCA and other metrics have been used be used to calculate environmental impact in manufacturing, such as Ecotax [11], Ecovalue08 [10], Ecoindicator-99 [12], Ecoinvent 3 [13], OEEE [14], etc.; each of which has unique advantages and disadvantages [15]. However, reducing the environmental externalities are not enough for companies: it is necessary to promote value creation and produce economic value [16]. That is, improve efficiency and performance. Efficiency improvement can be reached by the elimination of wastes, the reduction of costs, and the improvement of efficiency through lean strategies [5]. Consequently, literature indicates that by implementing green and lean practices simultaneously the organizations can enhance their business performance while creating environmental, social and economic benefits [17].

2.2 Lean Manufacturing

Lean is considered the most influential new paradigm in manufacturing [18]. Lean manufacturing improves the competitiveness of organizations by reducing inventories and lead-times, and improving productivity and quality [1]. Lean manufacturing has not only been in line with classical organizational objectives such as profitability and efficiency but also objectives that include customer satisfaction, and quality [18]. However, the benefits of lean are even more extensive, including strategic direction and strategic operational improvements that cover a broader range of sustainability issues, as the environmental impacts [19].

The adoption of improvement philosophies and a lean culture offers new opportunities for improving the quality and service in sectors associated with several industrial sectors [20]. Lean tools have recorded significant successes resulting in a worldwide and across sectors recognition including both products and services. There are business sectors and their corresponding supply chain networks that the application of lean thinking techniques seems to be very promising [21].

In the manufacturing sector, companies base some of their improvement activities on optimizing a well-known rate called OEE (Overall Equipment Efficiency). Although this rate was originally developed in the maintenance area, it spread to the whole company and is used to optimize operational activities.

2.3 Overall Equipment Effectiveness

Using OEE for performance measurement purposes is common in manufacturing across the globe, and the scientific community has paid attention to both lean (of which OEE is considered to be a part) and OEE [22].

The original definition of OEE developed by Nakajima [23] comprises the six big losses divided into the three categories: quality (Q), performance (P) and availability (A) (see Fig. 1). The relationship between the number of units produced and the number of units produced that meet specifications is the *quality rate* (Q). The *performance rate* (P) indicates the actual deviation in time from ideal cycle time. *Availability* (A) measures the total time that the system is not operating because of breakdown, set-up and adjustment, and other stoppages. It indicates the ratio of actual operating time to the planned time available [24].

However, successes and failures of lean manufacturing and other similar initiatives are highly context-dependent. Failures in lean implementation tend to stem from a lack of understanding of sector-specific contextual factors and organizational factors [25]. Consequently, it is necessary and valuable to develop the appropriate decision support algorithms, tools, metrics and a standard environmental impact assessment tool according to the context of the manufacturing industry (e.g., supply chain, logistics, etc.).

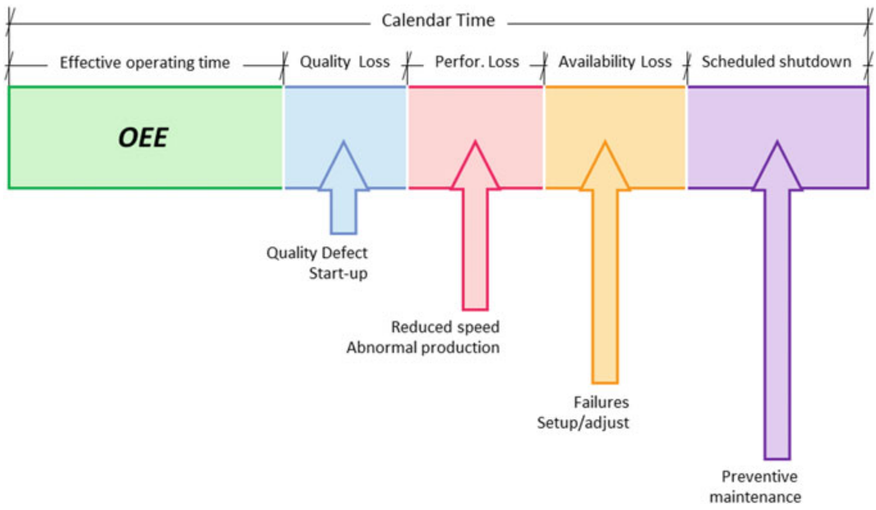


Fig. 1 OEE timeline. *Source* the authors

3 OGP: Overall Greenness Performance

Industrial performance metrics are shifting from economic-centric performance measures to those of sustainability [26]. This requires not only baseline information and temporal and spatial patterns to evaluate the status of environmental health and well-being [27], but also the design of metrics (indicators) by academy, the participation and validation by industry and the development of regulations by governments. Thus, one important step is for researchers to develop methodologies that use correct metrics, as metrics record quantifiable trends in observable phenomena in a simple and useful manner [28].

The OGP is a hierarchy of metrics that measure how environmentally efficient a manufacturing operation is realized according to VA (Value adding) processes that the consumer is willing to pay for. To do so, our indicator takes into account four different components (see Fig. 2). Note that each component measures consumption of resources (e.g., energy, water, etc.) and/or greenhouse gas (GHG) emissions (e.g., CO_2). The definition of measured resources/emissions should be previously defined by decision-makers. The four components are:

1. *Company context*: the environmental performance of a company greatly depends on where it is located. These components measure two different aspects. On the one hand, the local environmental legislation, which sets requirements that affect how the company performs certain processes. On the other hand, the environmental culture of working people, which sets the level of workers' environmental commitment.

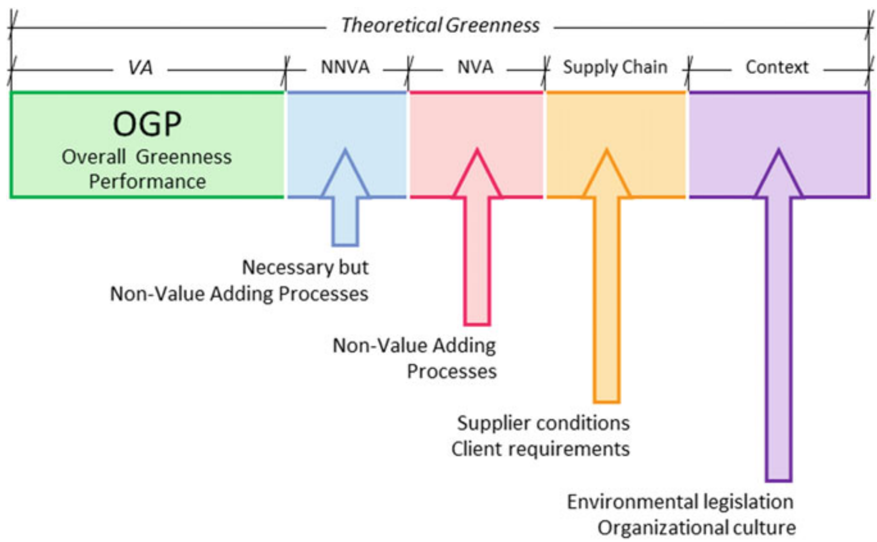


Fig. 2 OGP timeline. *Source* the authors

2. *Supply chain*: customers and suppliers also affect the environmental efficiency of a company; for example, by setting new restrictions on packaging, transportation, etc. This component measures those special requirements or conditions in the supply chain.
3. *NVA*: This component measures consumption/emission of the NVA (Non-Value Adding) processes, which can be easily removed from the process. Clearly, this is where the company can best improve its environmental impact.
4. *NNVA*: This component measures consumption/emission of the NNVA (Necessary but Non-Value Adding) processes. These processes are needed to connect different parts of the process (e.g., transport, inspection) and can also be removed by using the correct tools (e.g., SMED, TPM, etc.).

Similarly to OEE, our indicator estimates several environmental performances, from theoretical to "real" greenness. An analogy between concepts in OEE (see Fig. 1) and the concepts of the proposed metric (see Fig. 2) are presented in detail in Table 1.

Finally, a practical example of the proposed tool is presented in Table 2 and Fig. 3. A theoretical analysis of these results could be useful for a better understanding of the proposed method. Note that [kWh] of energy consumption and [kg] of CO_2 emissions are selected as the comparative measures.

It is important to mention that our approach can use different metrics as water consumption, waste, etc. As mentioned before, the definition of measured resources/emissions should be previously defined by decision-makers. This definition may be mainly limited by the available technology (or resources to acquire it) for measurement and control. In certain cases, emissions and/or consumptions can be estimated using different approaches.

Table 1 Analogies between OEE and OGP

OEE	OGP
Calendar time	**Theoretical greenness**
Scheduled shutdown Preventive maintenance	**Context** Environmental legislation and organizational culture
Availability loss Defects and start-up	**Supply chain** Special conditions/requirements
Performance loss Reduced speed Abnormal production	**NVA** Non-value adding processes
Quality loss Process failures Equipment breakdown Setup/adjust	**NNVA** Necessary but non-value adding processes
Effective operating time	**VA** Value adding processes

Table 2 Practical example—environmental company's performance

Component	Monthly energy consumption (kWh)	Monthly CO_2 emissions (kg)
VA	30	10
NNVA	10	5
NVA	5	10
Supply chain	2	5
Context	0.5	1
Total	47.5	31

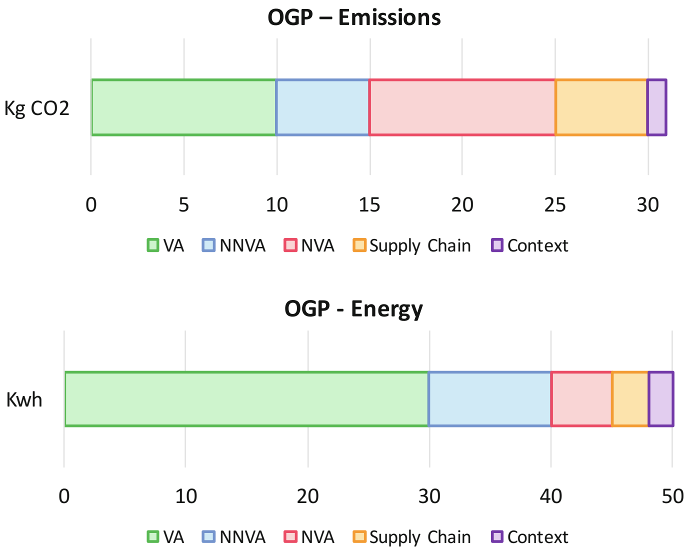


Fig. 3 OGP timeline of practical example

For example, aggregating information of fuel consumed to calculate the emissions of CO_2 [29] or the emissions created by the production of the energy and the emissions created by the consumption of the energy itself [8].

In this example, the value adding activities report 33% and 60% of CO_2 emissions and energy consumption, respectively. It is expected that VA activities (which are represented by the OGP) have a greater proportion of both resource consumption and a GHG emissions. However, there is a clear symptom of alarm that indicates that company must re-evaluate its production process regarding energy consumption, as 67% of CO_2 emissions do not correspond to VA activities.

On the other hand, NNVA and NVA activities represent potential improvement opportunities. As mentioned before, these activities could be easily removed from the process using the correctly tools (e.g., SMED, TPM, etc.). Furthermore, several projects, as LIFE MCUBO, aim to engage industry in efficient resource management through new methodologies, models and wireless technologies allowing them to simultaneously measure the productive and environmental efficiency of manufacturing processes [30].

In addition, the requirements and conditions of the supply chain represent 16% and 6% of CO_2 emissions and energy consumption, respectively. If significant consumption of resources and emissions are generated in transport activities, the company could focus on misleading collaborative transport strategies and/or distribution optimization models [31, 32].

Finally, two additional highlights can be derived. First, it is necessary to compare whether total resource consumption and GHG emissions are below regulatory legislation, including company's context component. And second, regarding to this context concept, it is important to evaluate the percentage of people or processes that are not yet synchronized with environmental development and therefore affect the overall performance of the company.

4 Conclusions

The Overall Greenness Performance (OGP) metric is presented in this paper as a tool for the assessment of environmental performance of companies. This metric allows companies to quantify sustainability practices in their organization in order to include it as a criterion in business decision-making. OGP metric could rapidly show the critical consumption/emission of several components (i.e., company context, supply chain and NVA, NNVA and VA activities) in order to devise and implement solutions that increase company performance.

However, as environmental efficiency should not be treated independently of productive efficiency, the consumption of energy, water, etc. should be monitored simultaneously with the production. For example, an increase in water and energy consumption is not obtrusive when production increases. On the other hand, it could be critical if this increase occurs in cleaning processes, a task that is necessary but does not add value to the product. Several correctly actions could be derived after the analysis of the proposed metrics and other tools as (i.e., SMED, optimization models, training, etc.) could be implemented to improve the critical areas detected.

Finally, an interesting line for future research may concern the design of easy-implantation and/or low-cost technology for the simultaneously measurement of production and energy (or resources) consumption in order to effectively implement this or similar approaches.

Acknowledgements The work presented in this paper was carried out within the framework of the LIFE MCUBO research project funded by the European Union (LIFE15 ENV/ES/000379), through the LIFE program. In addition, the authors would like to acknowledge the support received from AdP, the Special Patrimonial Fund at Universidad de La Sabana and the doctoral grant from TECNUN Escuela de Ingenieros, Universidad de Navarra.

References

1. Santos, J., Wysk, R., Torres, J.: Improving Production with Lean Thinking. Wiley, Hoboken (2014)
2. Alayón, C., Säfsten, K., Johansson, G.: Conceptual sustainable production principles in practice: do they reflect what companies do? J. Clean. Prod. **141**, 693–701 (2017)
3. Madan Shankar, K., Kannan, D., Udhaya Kumar, P.: Analyzing sustainable manufacturing practices—a case study in Indian context. J. Clean. Prod. (2017) (Article in Press). https://doi.org/10.1016/j.jclepro.2017.05.097
4. Esmaeilian, B., Behdad, S., Wang, B.: The evolution and future of manufacturing: a review. J. Manuf. Syst. **39**, 79–100 (2016)
5. Carvalho, H., Govindan, K., Azevedo, S.G., Cruz-Machado, V.: Modelling green and lean supply chains: an eco-efficiency perspective. Resour. Conserv. Recycl. **120**, 75–87 (2017)
6. Simpson, D.F., Power, D.: Use the supply relationship to develop lean and green suppliers. Supply Chain Manag. **10**(1), 60–68 (2005)
7. Montabon, F., Sroufe, R., Narasimhan, R.: An examination of corporate reporting, environmental management practices and firm performance. J. Oper. Manag. **25**(5), 998–1014 (2007)
8. Muñoz-Villamizar, A., Santos, J., Viles, E., Ormazabal, M.: Manufacturing and environmental practices in the Spanish context. J. Clean. Prod. (2017) (In Review)
9. Brundtland, G.: Our Common Future: Report of the 1987 World Commission on Environment and Development. Oxford University Press, UK (1987)
10. Ahlroth, S., Finnveden, G.: Ecovalue08-A new valuation set for environmental systems analysis tools. J. Clean. Prod. **19**(17–18), 1994–2003 (2011)
11. Finnveden, G., Eldh, P., Johansson, J.: Weighting in LCA based on ecotaxes: development of a mid-point method and experiences from case studies. Int. J. Life Cycle Assess. **11**(1), 81–88 (2006)
12. Bajdur, W.M., Henclik, A., Skowron-Grabowska, B., Iwaszczuk, N.: LCA application in the assessment of new technologies of industrial effluents treatment. Desalin. Water Treat. **57**(3) (2016)
13. Weidema, B.P., Bauer, C., Hischier, R., Mutel, C., Nemecek, T., Reinhard, J., Vadenbo, C.O., Wernet, G.: Overview and Methodology. In: Data Quality Guideline for the Ecoinvent Database Version vol. 3. The Ecoinvent Centre, St. Gallen, Switzerland (2013)
14. Domingo, R., Aguado, S.: Overall environmental equipment effectiveness as a metric of a lean and green manufacturing system. Sustainability **7**(7), 9031–9047 (2015)
15. Hertin, J., Jacob, K., Pesch, U., Pacchi, C.: The production and use of knowledge in regulatory impact assessment—an empirical analysis. For. Policy Econ. **11**(5–6), 413–421 (2009)
16. Vachon, S., Klassen, R.D.: Environmental management and manufacturing performance: the role of collaboration in the supply chain. Int. J. Prod. Econ. **111**(2), 299–315 (2008)
17. Cherrafi, A., Elfezazi, S., Garza-Reyes, J.A., Benhida, K., Mokhlis, A.: Barriers in green lean implementation: a combined systematic literature review and interpretive structural modelling approach. Prod. Planning Control 1–14 (2017) (Article in Press). https://doi.org/10.1080/09537287.2017.1324184
18. Forrester, P.L., Shimizu, U.K., Soriano-Meier, H., Garza-Reyes, J.A., Basso, L.F.C.: Lean production, market share and value creation in the agricultural machinery sector in Brazil. J. Manuf. Technol. Manage. **21**(7), 853–871 (2010)
19. Piercy, N., Rich, N.: The relationship between lean operations and sustainable operations. Int. J. Oper. Prod. Manage. **35**(2), 282–315 (2015)
20. Wang, B.: Analysis of efficiency of lean production implemented in multi-national optic enterprises. Int. J. Technol. Manage. **43**(4), 304–319 (2008)

21. Folinas, D., Aidonis, D., Malindretos, G., Voulgarakis, N., Triantafillou, D.: Greening the agrifood supply chain with lean thinking practices. Int. J. Agric. Resour. Gov. Ecol. **10**(2), 129–145 (2014)
22. Andersson, C., Bellgran, M.: On the complexity of using performance measures: Enhancing sustained production improvement capability by combining OEE and productivity. J. Manuf. Syst. **35**, 144–154 (2015)
23. Nakajima, S.: An Introduction to TPM. Productivity Press, Portland, OR (1988)
24. Jonsson, P., Lesshammar, M.: Evaluation and improvement of manufacturing performance measurement systems—the role of OEE. Int. J. Oper. Prod. Manage. **19**(1), 55–78 (1999)
25. Dora, M., Gellynck, X.: House of lean for food processing SMEs. Trends Food Sci. Technol. **44**(2), 272–281 (2015)
26. Abdul-Rashid, S., Sakundarini, N., Raja Ghazilla, R., Thurasamy, R.: The impact of sustainable manufacturing practices on sustainability performance. Int. J. Oper. Prod. Manage. **37**(2), 182–204 (2017)
27. Baird, R.C.: On sustainability, estuaries, and ecosystem restoration: the art of the practical. Restor. Ecol. **13**(1), 154–158 (2005)
28. Santos, J., Muñoz-Villamizar, A., Viles, E., Ormazabal, M.: Using a serious game for the practical learning of productive and environmental efficiencies, XXI Congreso de Ingeniería de Organización (CIO 17) (2017)
29. Sahu, S.K., Narayanan, K.: O_2 emission and firm heterogeneity: a study of metals and metal-based industries. In India. Int. J. Energy Technol. Policy **9**(2), 110–127 (2013)
30. LIFE MCUBO: About LIFE MCUBO. http://lifemcubo.eu/en/about-life-mcubo/ (2016). Accessed 2 July 2017
31. Montoya-Torres, J.R., Muñoz-Villamizar, A., Vega-Mejía, C.A.: On the impact of collaborative strategies for goods delivery in city logistic. Prod. Plann. Control **27**(6), 443–455 (2016)
32. Garza-Reyes, J.A.: Lean and green-a systematic review of the state of the art literature. J. Clean. Prod. **102**, 18–29 (2015)

A Maturity Framework for Operational Resilience and Its Application to Production Control

Duncan McFarlane, Rengarajan Srinivasan, Alena Puchkova, Alan Thorne and Alexandra Brintrup

Abstract This paper is concerned with resilience and its role in the operations of industrial processes. We refer here to operational resilience as the ability of an (industrial) operation to respond and recover in the face of unexpected or uncontrollable disruptions. The aims of this paper are to provide a common framework for examining the different challenges associated with assessing and improving operation resilience (b) Identify a set of levels for assessing operational resilience capabilities which can enable the positioning and comparison of initiatives taken to assess and improve it. (c) To illustrate the use of the operational resilience framework in the case of a laboratory forming and assembly operation.

Keywords Operational resilience · Industrial process · Production control

1 Introduction

This paper is concerned with resilience and its role in the operations of industrial processes. By *operational resilience* in this context we refer to: *The ability of an (industrial) operation to respond and recover in the face of unexpected or uncontrollable disruptions.*

D. McFarlane (✉) · R. Srinivasan · A. Puchkova · A. Thorne · A. Brintrup
Institute for Manufacturing, University of Cambridge, 17 Charles Babbage Road, Cambridge CB3 0FS, UK
e-mail: dcm@eng.cam.ac.uk

R. Srinivasan
e-mail: rs538@eng.cam.ac.uk

A. Puchkova
e-mail: ap823@eng.cam.ac.uk

A. Thorne
e-mail: ajt@eng.cam.ac.uk

A. Brintrup
e-mail: ab702@eng.cam.ac.uk

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_5

In our research we are particularly interested in the way in which resilience considerations can influence the decision-making processes that guide operations, but in this paper we will also note that resilience, and particularly the improvement of resilience, is dependent on preparations made in anticipation of disruptions and the gathering of information that can inform the onset of disruptions.

Resilience is something we work into our everyday lives—either deliberately or instinctively we will often take actions in order to prevent disruptions to a task or to limit the impact of one. If we are driving across a busy town, we might choose a route that is less likely to be jammed in heavy traffic, even if the distance is a little further. Our bodies build reserve supplies of anti-bodies in order to be able to fight an infection if we cut ourselves. We buy more tins of food than we need immediately in case additional visitors arrive or it turns out to be difficult to get the shop in the next few days.

In all of these cases we are:

(i) *being aware* of signs that a disruption is likely to occur [traffic banking up, pain signals, phone call from visitors];
(ii) *making preparations* to reduce either the chance of a disruption occurring [e.g. traffic delay] or to reduce the impact of a disruption on the task [e.g. plant dying, visitors arriving unexpectedly]. In each case, there is a cost and/or a reduction in performance [longer travel distance, more energy and food required to build antibodies, greater shopping cost and storage space] associated with this preparation;
(iii) once there is a sign that a disruption will occur, *managing* the onset of the disruption. This may involve doing nothing different [e.g. staying on the route if it is the best route in traffic] or adapting a plan [e.g. deciding to expand the soup being cooked to accommodate more diners, injecting antibiotics to supplement antibodies to the wound site] where the decision made is dependent on …;
(iv) assessing the *capabilities physically available for responding* to the disruption [e.g. alternative routes through street, larger saucepan, health of the person injured].

Achieving resilience to disruptions in an industrial context requires similar challenges—if a little more complex at times—and these four interlinked areas form the basis of the framework proposed in this paper for examining operational resilience.

The aims of this paper are to

a. Provide a common framework for examining the different challenges associated with assessing and improving operation resilience.
b. Identify a set of levels for assessing operational resilience capabilities which can enable the positioning and comparison of initiatives taken to assess and improve it.

c. To illustrate the use of the operational resilience framework in the case of a laboratory forming and assembly operation developed in conjunction with an industrial aerospace collaborator.

We note again that the scope of this paper is restricted to operations and also comment that the developed framework is most appropriate for day to day disruptions which limit operational capacity rather than *disasters* which lead to operational stoppages for significant lengths of time.

2 Review

The review material presented here encompasses work on resilience of industrial supply chain operations and of individual production operations. Although we don't distinguish directly between the two areas, we note that often the emphasis in the supply chain literature is on qualitative assessment and analysis of resilience while in the production literature the focus is more on quantitative methods improving resilience.

2.1 Definitions and Metrics

By industrial operations in this paper we mean the day to day processes associated with the making, moving and storage of raw materials, parts and products required to fulfil a customer order. The definition of resilience of industrial operations in Sect. 1 draws directly from this. We note that there are numerous definitions of resilience used in the academic literature. For example, in the supply chain literature, resilience is defined as "the ability of the system to return to its original state or move to a new, more desirable state after being disturbed" [1]. The essence of most of the definitions is associated with the response and recovery in the face of a disruption.

2.2 Examining and Assessing Resilience

Related to the definition of resilience is the notion of impact of a disruption which is generally viewed as the lost operational capacity over the time taken for recovery to be achieved. (e.g. [2, 3]). Impact is often the focus of an analysis of the nature and effects of disruptions on industrial operations. And in further research the notions of resilience capabilities have been addressed in different ways [3]. Numerous approaches to assess supply chain resilience have been proposed in the literature.

In general, these resilience strategies are classified as *mitigation* (pre-disruption) or *contingency* (post-disruption) strategies and include redundancy, flexibility, supply chain situation awareness, vulnerability assessment, supply chain collaboration and reengineering [1, 2].

2.3 Strategies for Managing Resilience

The *improvement of resilience* covers both the ability to reduce the incidence of unexpected disruptions and the ability to more effectively absorb or adapt those disruptions which do occur. Strategies for improving resilience generally involve making the most effective use of capabilities available or increasing levels of capabilities. For example, in [3] methods for quantifying capabilities have been categorised and the notion of potential capabilities and the ability to use that potential were discussed.

One of the major focus areas in improving production operational resilience is in developing different inventory mitigation strategies and contingency re-routing. Existing works model the production systems as an interconnected network of processes and mathematical tools are employed to determine different inventory control policies during disruptions [4, 5]. Additionally, real-time management and control of production systems during disruptions have also been studied considerably.

The application of intelligent control systems that can identify and manage disruptions have been proposed by employing concepts from multi-agent systems and holonic manufacturing. In such systems, the production entities are represented as agents or holons which can make distributed decisions and disruptions are managed dynamically by cooperating with other agents in the systems [6–8].

2.4 Challenges in Developing Resilience/Managing Disruptions

While there has been extensive research in the area of resilience generally, there remain some significant challenges: Classification categorising types of disruptions is given below:

- Determining the justifiable level of expenditure on operation resilience, especially where such an investment may be contrary to more common lean manufacturing practices [5].
- Determining where best to invest in resilience prevention or reactivity in the face of the disruption.
- Quantifying not only the initial impact of a disruption but also the propagation or the so-called *knock on effect* which is often more significant than the disruption itself.

- Articulating levels of operational resilience which can support a resilience improvement development path.

More broadly, there is a need for a framework which enables each of these issues to be put into the context of the whole challenge of managing a disruption from awareness to response as outlined in Sect. 1.

3 A Framework for Managing and Improving Resilience

In response to the challenges outlined in the previous section we introduce in this section a simple framework for capturing different dimensions of operational resilience.

3.1 Aims and Rationale

The aim of this framework is that of enabling different areas of resilience management to be clearly articulated and classified. It should also support an approach to measuring the extent to which a specific operation can said to be resilient overall or with respect to a class of disruptions—potential or actual.

3.2 Framework Overview

The framework proposed—the Operational Resilience Framework—is intended to capture aspects of the key questions that would be practically asked when ensuring a reasonable degree of resilience is in place for a particular operation. The four dimensions are:

1. *Awareness*: What is known about previous or future potential disruptions?
2. *Preparation*: What preparations can be put in place in anticipation of potential disruptions?
3. *Management*: What information processing/decision-making needs to take place to manage the disruption?
4. *Response*: What physical actions can be taken in response to the disruption?

Referring to Fig. 1 it can be seen that these separate dimensions reflect also the chronological steps associated with dealing with the onset of disruption, *awareness* and *preparation* representing actions prior to the onset of a disruption and *management* and *response* following the incidence of one or more disruptions.

The dashed arrow is intended to denote the learning from disruption to disruption which feeds back into awareness and preparation stages.

3.3 Details of Operational Resilience Dimensions

In this section, we expand each of the dimensions to indicate the typical information gathered, decisions made and actions. These are the result of practical studies of disruption management over a twenty-year period in manufacturing, logistics, service and aviation sectors.

Table 1 provides what is effectively a *check list* for operational resilience. The level to which each item is addressed will be addressed in the following section.

4 Resilience Maturity for Production Operations

Following on from the detailed description of the different dimensions of operational resilience is the intent to be able to distinguish between different levels of development that might be present in the particular company's activities.

4.1 Why Measure Resilience Levels

The primary rationale for introducing resilience levels is to provide an organisation with a means of benchmarking its evolving ability to manage disruptions. Maturity

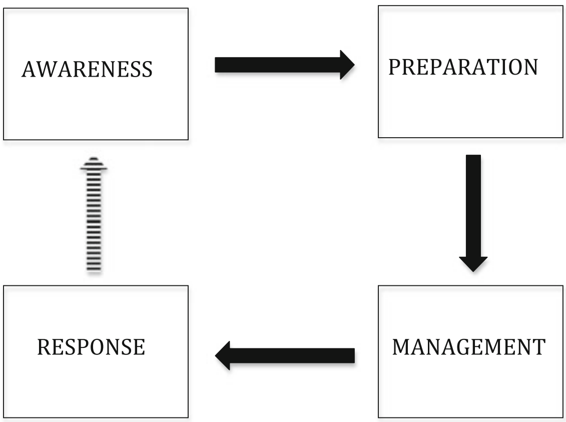


Fig. 1 Resilience or disruption management cycle

Table 1 Typical inputs/actions for each resilience dimension

Area	Issue	Typical information/decisions/ actions
Awareness	What is known about previous or future potential disruptions?	**Historical Information**: typical causes of delays in operation in question, impact of delays locally and on surrounding operations, frequency of typical types of delays
		Current Operational Information: disruption indicators, part/product tracking systems, equipment state monitoring
Prediction	What preparations can be put in place in anticipation of potential disruptions?	**Resources**: adding flexibilities to equipment, adding additional redundant capacity (human/ equipment), disruption training, alternative routes identified
		Schedules: adding appropriate slack to schedules on key equipment, operators, lines
		Parts: build appropriate buffers of raw materials, work in process, finished goods
Management	What information processing/ decision making needs to take place to manage the disruption?	**Information Processing**: Tracking of current system state and recent history, detection, location and diagnosis of disruptions
		Decision Making: Strategy to manoeuvre system into recovery state, schedule recalculation, determine and update instructions to resources (machines, lines, operators), determine and trigger replacement parts
Response	What physical actions can be taken in response to the disruption?	**Resources**: equipment start-up/shut down procedures, buffer building triggered, maintenance and repair, equipment/line reconfiguration, operator redeployment
		Schedules: updated following recalculation, order completion and delivery changes communicated
		Parts: replacement parts sourced, transported

modelling has been shown to be a useful tool for articulating the capability development of an organisation—allowing a measured, staged approach to be taken using objective measures and terminology (see for example [9, 10]).

Table 2 Maturity levels for the operational resilience framework

Level	Awareness	Prediction	Management	Response
0	None. No expectation of disruptions beyond normal process variations	No specific preparations. Set up of machines, schedules, buffers for normal operation only	Limited tracking of orders and parts. No abnormal event detection. Control systems designed to manage normal ops	Instinctive. Limited capabilities available than for standard repair, replace, restart. Disruptions are unexpected
1	General expectation of disruption types and their sources	Some additional allowances made in schedules, set up, buffers, equipment to accommodate disruptions	Unique tracking of parts and orders. Detection of some abnormal events (e.g. MCM, QA). Control able to exploit allowances	Equipment and other resources access spare capacity. Buffers triggered to replenish. Schedule repaired/updated. Part replacement enabled
2	Statistical expectation of type, level, distribution, frequency of disruptions over time	As per Level 1 but historical disruption patterns influence the way in which the different additional allowances are introduced	As per Level 1 but control strategy is designed to best support disruption patterns that have been identified	In addition to Level 1, specific adjustments to resources, parts/ WIP, schedules can be for particular disruption types
3	Real time updating of statistical expectations and predictions of disruptions using process, equipment and quality data	Real time data from process, equipment and quality etc. is used to regularly adapt the allowances made for likely disruptions	As per Level 2 but system is continuously adapting the disruption management control strategy based on any real time data, and using machine learning to optimise new logic	As per Level 2 but the response to the onset of a disruption is almost invisible as the production system continues to operate as "business as usual"

4.2 Levels of Industrial Resilience

The levels selected to support each of the resilience dimensions of the Operational Resilience Framework are based on the following approach:

- *Level 0*—little or no direct appreciation of disruptions within the day to day operations

- *Level 1*—appreciation in general and qualitative terms of the potential disruptions
- *Level 2*—a detailed quantitative understanding of disruptions and how to manage them
- *Level 3*—real time learning based approach to improving resilience

Table 2 provides the maturity framework for operational resilience which we note can be applied to an operation as a whole or to an operation with respect to one particular class of disruptions.

The case study in the following section will illustrate the use of the maturity levels in assessing the resilience of an existing operation and designing improvements.

5 Case Study: Production Resilience in the Control of a Laboratory Forming and Assembly Operation

5.1 Problem Description

Working with an industrial partner, a laboratory forming and assembly production system was developed so that the typical disruptions experienced within the partner's operations were reflected in this laboratory scale system. The intent was to mirror the operation's current resilience capabilities, develop an improvement strategy and then to migrate aspects of this strategy back into the partner's operational environment.

5.2 Operational Overview

Figure 2 shows the layout of the production system.

The production system manufactures a simple gearbox. The metal casing is machined by a 5-axis CNC machine and the plastic covers are made by a vacuum forming machine. Cell 1 is manufacturing cell, where the metal parts are machined and the plastic parts are formed. Cell 2 is the sub-assembly process where the metal top and the plastic covers are aggregated. Cell 3 is the final assembly cell associated with gear meshing and fastening operations. Material movement between cells are done by manually moving trolleys and also through an automated conveyor system using shuttles.

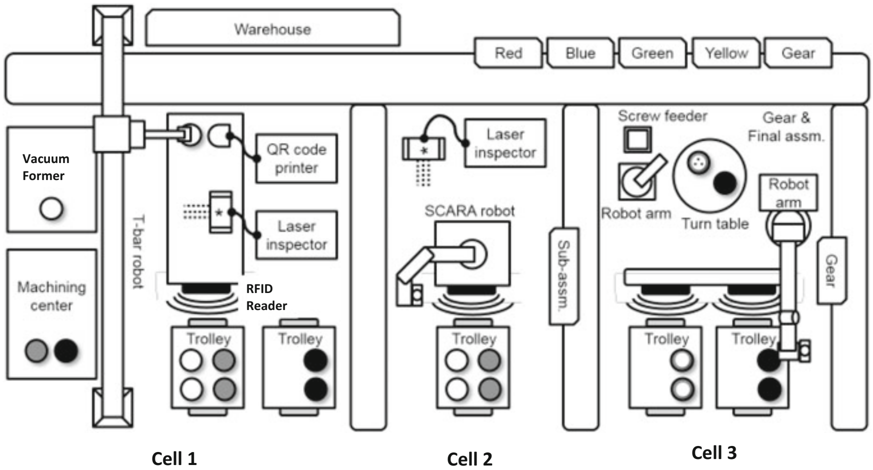


Fig. 2 Forming and assembly operation layout

5.3 Operational Resilience Maturity Levels

We now review the resilience maturity levels of the original operation, the improved operation and the first stage of migration of the improved solution back to the original partner operation.

(a) **Resilience of Original Operation**—referring to Fig. 3a (which depicts the Operational Maturity levels for the three stages of development), the understanding of the resilience of this operation initially was very limited. The operation had been developed along lean manufacturing lines with limited awareness of the potential onset of a disruption, little or no preparation and disruptions occurring were managed on an ad hoc basis.

(b) **Improved Operational Resilience Approach**—in order to address the limited ability of the system to withstand disruptions a number of significant improvements were introduced (refer to Fig. 3b):

- *Awareness*: key disruption types were identified—namely quality misses and equipment breakdowns—and this information fed into the preparation stage;
- *Preparation*: additional spare parts were added to the system to provide buffering in case of disruption. (These developments are reported in detail in [5]);
- *Management*: An agent based production control system was introduced to explicitly and implicitly add the ability to continue to operate in the face of disruptions of the classes identified. In addition, a part and order tracking system was introduced to provide up to date status information to the control system. (These developments are reported in detail in [11]);

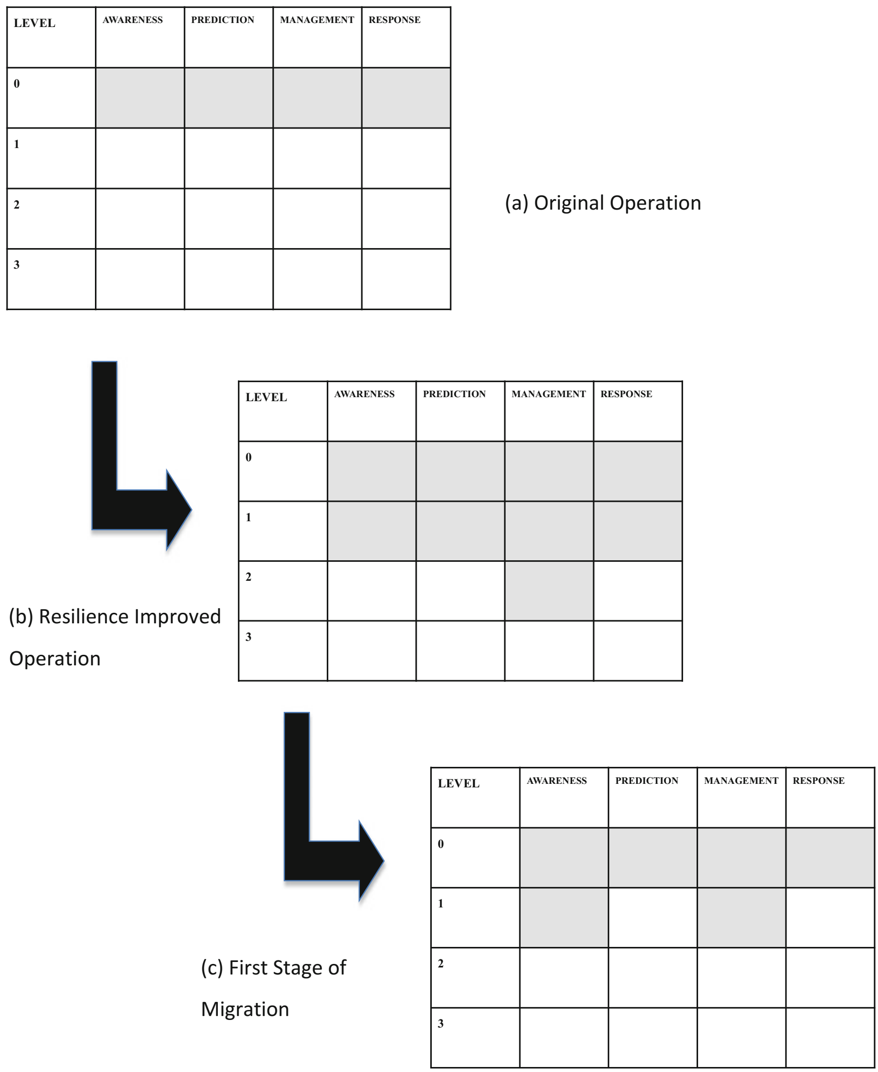


Fig. 3 Resilience maturity for case study operation

- *Response*: the ability to interact with additional storage areas was added so that the system could flexibly access spare parts. Alternate assembly routes were also made available should break down of certain equipment occur.

(c) **Migration to the Partner Operations**—In 2016 a migration programme was instigated to begin to introduce the laboratory developments back into the operations of the industrial partner. As well as a more detailed analysis of the types of disruptions occurring and their frequency and impact, the initial focus of this work has been to develop aspects of the part and order tracking system discussed above and is illustrated in the maturity framework in Fig. 3c.

6 Conclusions

This paper has presented a simple framework for capturing different dimensions and development stages of operational resilience. The aim of such a framework is to provide a systematic process for improving resilience and current research is exploring its use in manufacturing, logistics and aviation case studies.

The particular production control case study presented here has been the subject of a range of resilience improvement initiatives and the framework has been helpful in helping to understand all of these initiatives in a single context.

References

1. Christopher, M., Peck, H.: Building the resilient supply chain. Int. J. Logistics Manag. **15**(2), 1–14 (2004)
2. Spiegler, V.L., Naim, M.M., Wikner, J.: A control engineering approach to the assessment of supply chain resilience. Int. J. Prod. Res. **50**(21), 6162–6187 (2012)
3. Matson, J., McFarlane, D.: Assessing the responsiveness of existing production operations. Int. J. Oper. Prod. Manag. **19**(8), 765–784 (1999)
4. Hu, Y., Li, J., Holloway, L.: Resilient control for serial manufacturing networks with advance notice of disruptions. IEEE Trans. Syst. Man Cybern. **43**, 98–114 (2013)
5. Puchkova, A., Srinivasan, R., McFarlane, D., Thorne, A.: Towards lean and resilient production, pp. 2387–2392. INCOM, Ottawa, Canada (2015)
6. Bussmann, S., Wooldridge, M.: On the identification of agents in the design of production control systems. In: Ciancarini, P., Wooldridge, M. J. (eds.) Agent-Oriented Software Engineering, pp. 141–162. Springer, Berlin, Heidelberg (2001)
7. Jennings, N.R., Bussmann, S.: Agent-based control system. IEEE Control Syst. Mag. (2003)
8. Leitao, P.: Agent-based distributed manufacturing control: a state-of-art survey. Eng. Appl. Artif. Intell. **22**, 979–991 (2009)
9. Maasouman, M.A., Demirli, K.: Development of a lean maturity model for operational level planning. Int. J. Adv. Manuf. Technol. **83**(5), 1171–1188 (2016)
10. Li, H., Ji, Y., Gu, X., Bao, Z., Qi, G.: A universal enterprise manufacturing services maturity model: a case study in a Chinese company. Int. J. Comput. Integr. Manuf. **27**(5) (2014)
11. Srinivasan, R., McFarlane, D., Thorne, A.: Identifying the requirements for resilient production control systems. In: Service Orientation in Holonic and Multi-Agent Manufacturing, pp. 125–134. Springer International Publishing (2016)

A Case Study of Intelligent Manufacturing Control Based on Multi-agents System to Deal with Batching and Sequencing on Rework Context

Emmanuel Zimmermann, Hind Bril El Haouzi, Philippe Thomas, Rémi Pannequin, Mélanie Noyel and André Thomas

Abstract Nowadays complex control systems are rising, and especially hybrid control architectures which are developed to face manufacturing control challenges that occur with the last industrial revolution and the emerging of industry 4.0. This work presents an application, on a testing platform, of a scheduling algorithm with multi-criteria objectives developed for the Acta-Mobilier company facing high rework rate. This algorithm will inscribe itself in a hybrid control system based on smart entities. The main objective is to validate the contribution of the proposed algorithm in a disturbed environment. The platform, implemented with a multi-agent system, allows measuring the reliability of the proposed algorithm used for a complex system in the particular case of high rework rate.

Keywords Hybrid control system • Intelligent manufacturing control
Viable system model • Multi-agents platform

E. Zimmermann (✉) · H. B. E. Haouzi · P. Thomas · R. Pannequin · A. Thomas
Université de Lorraine, CRAN, UMR 7039, Campus Sciences, BP 70239, 54506
Vandœuvre-lès-Nancy Cedex, France
e-mail: emmanuel.zimmermann@univ-lorraine.fr

H. B. E. Haouzi
e-mail: hind.el-haouzi@univ-lorraine.fr

P. Thomas
e-mail: philippe.thomas@univ-lorraine.fr

R. Pannequin
e-mail: remi.pannequin@univ-lorraine.fr

A. Thomas
e-mail: andre.thomas@univ-lorraine.fr

E. Zimmermann · M. Noyel
Acta-Mobilier, Parc d'activité Macherin Auxerre Nord, 89270 Moneteau, France
e-mail: mnoyel@acta-mobilier.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_6

1 Introduction

The growing trend of complex and customizable product presents many challenges: on one hand, the number of product types to manufacture is increasing constantly and needs efficient knowledge and information management. On another hand, the introduction of new products stimulated adapting and reconfiguring production tools frequently. This adaptation need led, beginning with the 90s, to numerous initiatives involving laboratories, universities and companies. The aim of these initiatives was to design and implement production systems of the future. The Intelligent Manufacturing System (IMS) Project [1] has been one of the main initiatives; the scope is to develop decentralized systems with high reliability to answer both exterior and interior disturbances. However, nowadays implementations are still insufficient due to inherited inertia in every company's paradigm changes. To evolve from hierarchical systems such as ERP or SCM systems to auto-organized and adaptive complex ones, we need deep changes in the organization's work, the company culture and also in technical infrastructures. That is why hybrid control solutions like the ones studied by [2, 3] since 2000 are explored.

Complex systems are generally composed by numerous autonomous entities with different adaptability levels and able to reveal important emerging phenomena. These phenomena cannot be easily induced from the knowledge about their components. This is why most of new complex system architectures are implemented using the multi agents system paradigm. The aim of this paper is to show a case study implementing a hybrid control architecture using the multi agent system framework and a laboratory test-based platform: TRACILOGIS. The implemented scenario is based on a real case study from the Acta-Mobilier company—a high-quality front manufacturer of kitchen and bathrooms. The company suffers from a high rework rate (upper than 40%) due to high quality requirements. A mass customisation strategy led the company to produce just in time but short delivery times and a large panel of different models increased the rework rate. These facts added to the customer demand's variability made difficult operational planning activities. This situation caused overdue front delivering and raised the production costs. Figure 1 represents, in macroscopic view, the production flow of Acta-Mobilier; each step represents a work centre and the grey arrows show the natural flow. The yellow arrows indicate possible rework loops which could appear; these loops can happen on any step of the process and are going to four different available destinations: *cutting* (front completely remade), *priming* (put a new primer painting layer to hide default), *lacquering* (put a new layer of lacquering to hide a default) or *rework* (specific workstation to correct small defects, not visible in the figure).

Three main operation flows are processed simultaneously:

- The brilliant lacquering flow: the only one passing on a polishing step
- The matt lacquering flow: send to packaging and shipment directly after lacquering
- The cement flow with a special cement application phase before lacquering

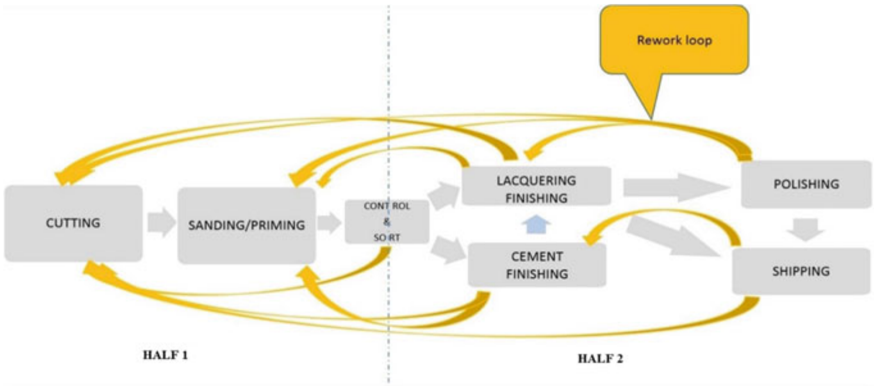


Fig. 1 Acta generic process

The diagram has been separated in two parts to mark that there is a significant functioning change in the production. In the first one, the fronts with the same physical characteristics are gathered and in the second part, they are grouped according to the finishing type or colour.

There are many levers to work on to assure efficiency of the production line, like load balancing between the three flows in the finishing work centre.

In section two, the problem is settled. In section three, a presentation of the test platform is given and the work carried out for the platform is described. Then, the experiment and its results are discussed. And in last section a conclusion is drawn and further enhancements are proposed for the platform.

2 Toward an Intelligent Manufacturing System

The implementation of such complex adaptable systems is mainly based on the DAI (Distributed Artificial Intelligence) paradigm and OOP (Object Oriented Programming). Without being exhaustive, some agent-based intelligent system architectures could be cited: YAMS [4], AARIA [5], METAMORPH [6], PROSA [7] (and especially its ARTI implementation) (D4U), InteRRaP [8], ADACOR [9], Pabadis Promise [10], FrMS [11], NEIMS [12], HCBA [13] or D-MAS [14].

Finding an efficient implementation of Hybrid Manufacturing System is still a topic of high attractiveness for agile adaptable systems researchers. Lately, works like ORCA-FMS [15], PROSIS [16] or ADACOR2 [17], Service—oriented Holonic Manufacturing Systems—SoHMS [18] were led.

Tools, criteria and model-based decisions remain extremely bound to the designer preferences, abilities and also to the studied problem specificities. CRAN investigated, through Herrera's works, a generic framework based on the VSM

(Viable System Model) applied in [19]. In these work, a recursive meta-model has been proposed to improve the efficiency of the production line, to reduce the delay and provide an adaptive schedule able to deal with reworks and machine failures. This control system is modelled with the cybernetic core described by Beer [20]. The proposed model is a hierarchical recursive system based on smart entities.

Figure 2 shows a representation of this model: on the lowest stage, there are the “Kernels” (minimal undividable groups/batches of fronts having exactly the same production range). They are considered as autonomous entities and are able to choose by themselves their passage order in a production batch. The next level represents these production batches, dynamics Kernels batches on one or more workstation. They are able to dismantle and build themselves in order to reach the local objectives of every workstation according to the schedule produced by the local optimizer. All the production batches of a given customer for a given week constitute a production order. Each production order is defined by its due date and its position in the production plan. They have to negotiate to schedule themselves according to data received from their “children” (kernels characteristics like colour, thickness, etc. …) and the environment (workstations breaks, weather conditions able to downgrade quality). To summarize, this chart explains that the system is made like a living being with autonomous entities on each level, having their own knowledge and sharing them with upper or lower entities according to the needs.

In [21] a local optimizer has been proposed in order to optimize the schedule on the cutting work center and responding to constraints of other work centers. This work corresponds to a part of the implementation of the area encircled in red on Fig. 2:

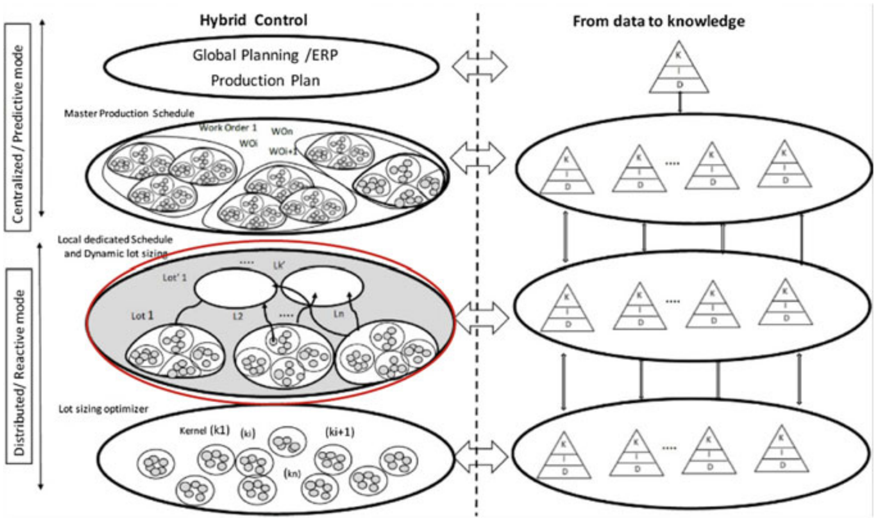


Fig. 2 Proposed meta-model of a hybrid viable system

- The goal in cutting work center is to minimize the number of tool changes and this way reduce the setup times.
- In finishing, finished product references have to be gathered as fast as possible and by minimizing the number of references worked simultaneously

This optimizer has been tested on a customer who orders only fronts from a single kind of flow: the cement one. Results seems meaningful but a relevant mathematical proof, impossible to obtain with the lack of retrievable data available in the company, would be appreciate.

It is non-trivial to test experimental solutions directly on the company production line due to the low action window available. That is why, the use of a test-based platform became necessary. CRAN own a platform called TRACILOGIS with the ability to test such experiments.

3 Implementation on TRACILOGIS Platform

3.1 The TRACILOGIS Platform

The TRACILOGIS test-based platform provides a test environment for many industrial scenarios on classification, identification and intelligent manufacturing control with different modes: centralized, distributed or hybrid (centralized and distributed). In addition, it allows implementing, testing and comparing various traceability and control techniques within the logistics chain, in particular for the wood industry. The physical system is composed of four automatons. They all handle their own part of the platform. The products are platter which can be marked with a point or a line on one of their border. They can receive between zero and four pallets and/or zero to four pastilles.

The global platform's physical processes are shown in Fig. 3:

- Automaton A handles a trail made up of two marking workstations. The workstations are placed sequentially but each one can be dodged thanks to referral (Fig. 3 area A).
- Automaton B manages a looping area which allows re-ordering products in a different way (Fig. 3 area B).
- Automaton C manages a sorting trail using a camera system to recognize pallet colours.
- Automaton D handles an assembly line: it adds the pallets and the pastilles on the platter according to the production range and then leads the products to one of the two exits (Fig. 3 area D).
- Both platters and pallets have RFID tags and readers are placed on each workstation and decision points.

The multi-agent system, developed with JADE, manages sending of all the actions to be realized to the automata. The transmission is triggered by the

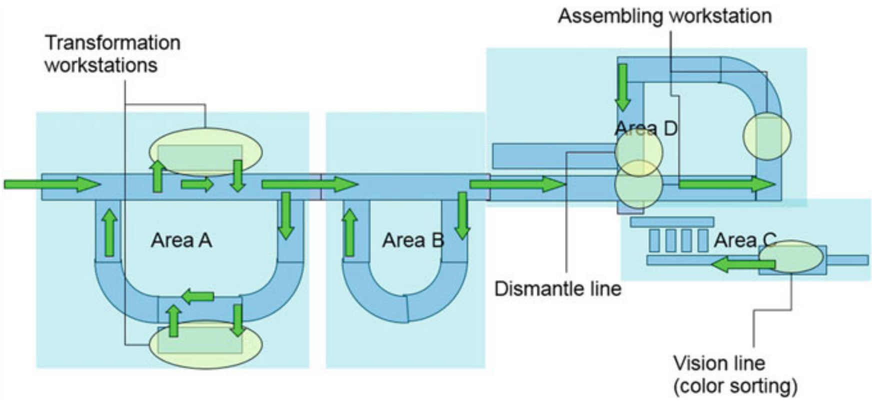


Fig. 3 TRACILOGIS flowchart

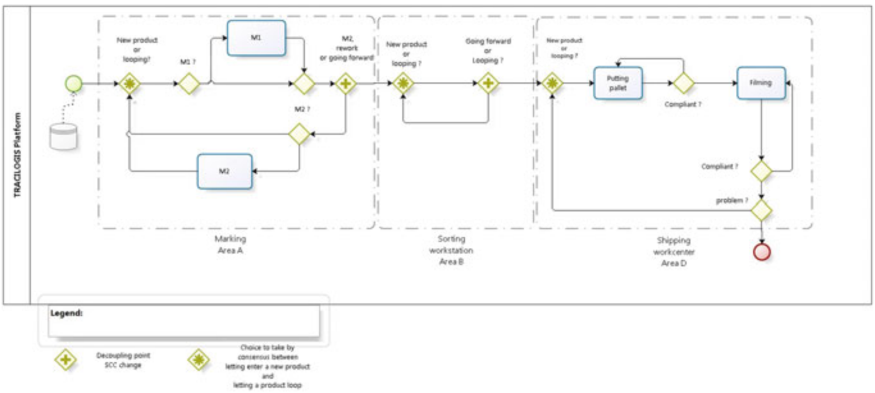


Fig. 4 TRACILOGIS UML diagram

behaviours implemented in the agents. Interactions are made by transmitting asynchronous messages.

In a multi-agent system, a (not necessarily) known number of agent are evolving. Each one is interacting with the others in order to reach a common goal. They can act also on a defined number of objects. A multi-agent system aims to maintain: a constant communication, the control and the knowledge organization between the agents (Fig. 4).

Initially agents were instantiated for the following type of actors on the platform:

- The *resource agents* are passive agents which wait for instructions from products.
- The *product agents* they know their objectives and their production range; they recalculate at every move the shortest path to their final destination.

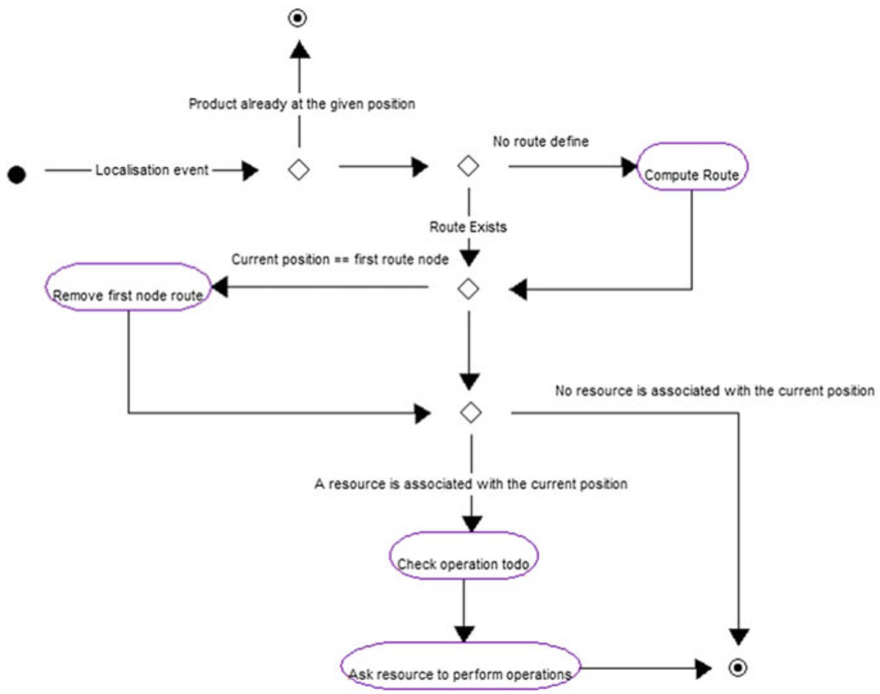


Fig. 5 Product agent UML

They are aware of their actual state but haven't awareness of the other product agents. The UML diagram of this agent is presented in Fig. 5.

- The *PLC* (Production Line Controller) *agents* manage all of the automaton actions of their area.

3.2 *Transposition of Acta-Mobilier Case Study on the Test-Based Platform*

The platform TRACILOGIS gives the opportunity to seek different kind of scenarios. The principal issue is to create study cases as near as possible of the industrial one in order to be relevant and make the platform evolve to realise these scenarios.

Moreover, the reliability of a multi-agent system in response to reworks disturbances could be analysed.

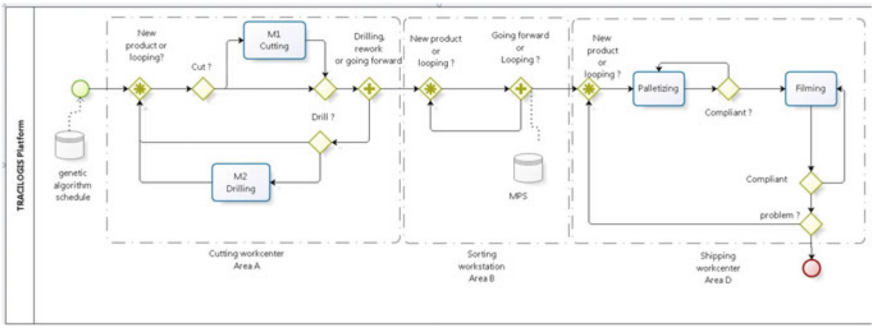


Fig. 6 Problem transposition from Acta to TRACILOGIS

- *Area A* of the platform could be associated to the cutting work-centre. The first machine can be seen as the normal flow of cutting and the second one as a special flow for particular model which is longer.
- *Area B* which is a simple loop workstation could be compared to the sorting phase of the company. This phase consists in verifying the completeness of the lots and regrouping them by colour Area B which is a simple loop workstation could be compared to the sorting phase of the company. This phase consists in verify the completeness of the lots and regroup them by colour. The worker will let the batch follows its way according to the production plan.
- *Area D* is compared to the finishing and shipment work-centers. For this experiment this area isn't really useful. Figure 6 resumes all these equivalences.

The first added implementation was the introduction of reworks. A random rule is placed in the resource agent behaviour to send a message to the product agent of a failure. When the product receives this answer, it will recalculate the shortest path to the machine.

A schedule agent has been introduced in order to assume that all the product which exit from area B are respecting the schedule otherwise the product will loop. The product agents communicate with this agent to know if they have the right to pass.

4 Experimental Results

A manufacturing of ten products has been implemented. Three different configurations were chosen: four products with a line made on machine M1 (Config. α of Fig. 7), three products with a point made on machine M1 (Config. β of Fig. 7) and three products with a line made on machine M2 (Config. γ of Fig. 7).

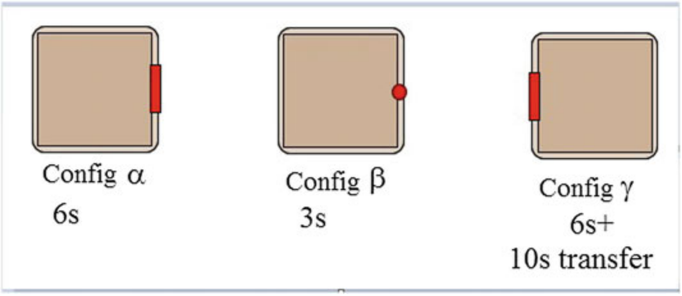


Fig. 7 Configuration list

- Config α corresponds to a line drawn on the right side of the product, operation available only on machine M1.
- Config β is a point drawn on the right side of the product, only available on machine M1 too.
- Config γ is a line plotted on the left side of the product; this operation is made on machine M2 and as shown on Fig. 6. In order to reach M2 the product should take the first looping road. That's why there is a 10 s transfer time.

The ten products are grouped in three customer orders: Order red: αβγ, Order green: αβγ and Order blue: αββγ. The three orders have the same due date.

To suit to Acta process, as soon as a product from a specific order goes through area B, the others products which don't belong to the same order will loop until the order is completed. To validate the advantages of the proposed solution, three different production schedule scenarios were tested:

- Each customer order is launched as a single batch without considering the setup optimization. Only the date is considered to schedule the jobs. This scenario has been named "to order schedule".
- The schedule is done based on the Due Date and operation change optimization (setup time). This scenario was named "DDS schedule".
- The schedule is done based on the results of the genetic algorithm (dealing with a trade-off between setup time and WIP. In this case the WIP is estimated by the number of loops). This one was named "GA schedule".

The following schedules have been established according to the three piloting rules described previously:

- Schedule 1: αγββγααββγ
- Schedule 2: αααββββγγγ
- Schedule 3: βββααγγγαβ

Table 1 Experiment results without reworks

	To order	DDS	GA
Number of loop	2	8	3
Makespan	5 min 08 s	4 min 44 s	4 min 56 s
Max FPRP	1	3	2
Operation changes	6	2	4

Table 1 shows the results considering four aspects:

(1) The number of loops needed to respect a specific piloting rule: only a product from a same order can go through area B, the others have to loop until all the products of the previous order have gone through area B. In other words, it corresponds to the measurement of the work in progress.
(2) The makespan.
(3) The maximal number of Finished Product References in Progress simultaneously (FPRP). In the schedules 1, 2, 3 products where coloured with three different colors, each of them representing a FPR; one of the main objective of the genetic algorithm proposed is to minimize the number of FPR worked simultaneously because the more distinct FPR are opened, the largest floor space to treat them is needed.
(4) The number of operation changes. Each operation change should be seen as a setup time.

The first part of the tests is done without reworks introduction. For these three experiments only two runs were realized (a normal and a control one) because no random ones were introduced in the system.

In this first part, it is to be noted that the "to order" schedule offers the less number of loops but also the maximal number of operation changes. This schedule is the longest because it doesn't take into account the optimisation of operations but allows working only a single FPR at a time.

Conversely, the minimizing setup times schedule gives the lowest number of changes but maximizes the number of loops and the number of FPR worked together.

The proposed GA has the advantage to provide a compromise solution of the others. Table 2 shows the results with 20% rework probability. For these ones, a set of ten runs were made and an average of the results is presented.

Table 2 Experiment results with 20% rework's probability

	To order	DDS	GA
Loop number	3	12	5
Makespan	9 min 38 s	9 min 13 s	9 min 21 s
Max FPRP	2	3	2
Operation changes	6 + 5	2 + 4	4 + 4

Fig. 8 Makespan measure dispersion

Comparing to the first part, the results are quite similar, except for the fact that all production times have more than doubled. However, the proposed GA solution seems to be robust to reworks.

The dispersion measure of the makespan is presented on Fig. 8. Through this diagram the “to order” scheduling is the one with the lowest dispersion but with results globally worse than the others. GA has the highest dispersion but also the best results; this could be explained by the number of iterations. This number is limited to maintain a reasonable computation amount. An increase of this number will probably allow to limit the dispersion of the solution and in the same time enhance the median. It will be interesting to evaluate the available computation time in order to maximize it utilization. Moreover, seeing that the junction between the median is not linear the scheduling strategy has a real impact on the makespan.

5 Conclusions

The average results highlight that the proposed solution is a good compromise to limit the number of operation changes and to limit the work in progress despite existing reworks. An interesting point not visible in the presented results, but noted however during the runs on the TRACILOGIS, is that the proposed algorithm, by grouping similar products and still paying attention to keep products from the same order as near as possible, allows switching products from an order to another in case of reworks. This fact isn’t really useful for a company like Acta because of its mass customisation strategy, but can still be relevant for companies with less distinct models. To be even more relevant, much more experiment should be made and with several distinct rework rates.

Further improvements of the platform and its multi-agent system would be to implement a consensus decision to choose the best solution between letting a new product enter or work on a rework. A batch agent must also be added to match with the notion of finished product reference and to maintain a communication between all products with the same reference. The next step to reach will be to recalculate the schedule dynamically each time a rework appears.

Acknowledgements The authors gratefully acknowledge the financial support of the CPER 2007–2013 Structuration du Pôle de Compétitivité Fibres Grand'Est (Competitiveness Fibre Cluster), through local (Conseil Général des Vosges), regional (Région Lorraine), national (DRRT and FNADT) and European (FEDER) funds used to implement the TRACILOGIS Platform.

References

1. Yoshikawa, H.: Manufacturing and the 21st century-intelligent manufacturing systems and the renaissance of the manufacturing industry. Technol. Forecast. Soc. Change 195–213 (1995)
2. Herrera, C., Belmokhtar, S., Thomas, A.: Viable system model approach for holonic product driven manufacturing systems. In: Proceedings of 1st Workshop on Service Orientation in Holonic and Multi Agent Manufacturing Control, SOHOMA (2011)
3. Herrera, C., Thomas, A., Belmokthar, S., Pannequin, R.: A viable system model for product-driven systems. In: Proceedings of the International Conference on Industrial Engineering and Systems Management, IESM (2011)
4. Van Dyke Parunak, H.: Manufacturing experience with the contract net. Distrib. Artif. Intell. 285–310 (1987)
5. Van Dyke Parunak, H., Baker, A., Clark, S.: The AARIA agent architecture: an example of requirements-driven agent-based system design. In: AGENTS'97 1st International Conference on Autonomous Agents (1997)
6. Maturana, F., Shen, W., Norrie, D.: MetaMorph: an adaptive agent-based architecture for intelligent manufacturing. Int. J. Prod. Res. 2159–2173 (1999)
7. Van Brussel, H., Wyns, J., Valckenaers, P., Bongareerts, L., Peeters, P.: Reference architecture for holonic manufacturing systems: PROSA. Comput. Ind. 255–274 (1998)
8. Fisher, K.: Agent-based design of holonic manufacturing systems. Robot Auton Syst. 3–13 (1999)
9. Leitao, P.: An agile and adaptive holonic architecture for manufacturing control. Ph.D. thesis, University of Porto (2004)
10. Wunsch, D., Bratukhin, A.: Multilevel order decomposition in distributed production. In 12th IEEE Conference on Emerging Technologies and Factory Automation EFTA'07 (2007)
11. Ryu, K., Jung, M.: Agent-based fractal architecture and modelling for developing distributed manufacturing systems. Int. J. Prod. Res. 4233–4255 (2003)
12. Gu, W., Tang, D., Zheng, K., Wang, L.: A neuroendocrine-inspired bionic manufacturing system. J. Syst. Sci. Syst. Eng. 275–293 (2011)
13. Chirn, J.L., McFarlane, D.C.: A holonic component-based approach to reconfigurable manufacturing control architecture. Database Expert Syst. Appl. 219–223 (2000)
14. Verstraete, P., Valckenaers, P., Van Brussel, H., Saint Germain, B., Hadeli, K., Van Belle, J.: Towards robust and efficient planning execution. Eng. Appl. Artif. Intell. **21**(3), 304–314 (2008)

15. Pach, C., Berger, T.: Bonte, T., et al.: ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. Comput. Ind. **65**(4), 706–720 (2014)
16. Pujo, P., Broissin, N., Ounnar, F.: PROSIS: an isoarchic structure for HMS control. Eng. Appl. Artif. Intell. **22**(7), 1034–1045 (2009)
17. Debnathm, S., Qin, J., Bahrani, B., et al.: Operation, control, and applications of the modular multilevel converter: a review. IEEE Trans. Power Electron. **30**(1), 37–53 (2015)
18. Quintanilla, F.G., Cardin, O., Castagna, P.: Product specification for flexible workflow orchestrations in service oriented holonic manufacturing systems. 177–193 (2014)
19. Zimmermann, E., Bril El Haouzi, H., Thomas, P., Thomas, A., Noyel, M.: A hybrid manufacturing control based on smart lots in a disrupted industrial context. In: Proceedings of IFAC World Congress (2017)
20. Stafford, B.: The viable system model: its provenance, development, methodology and pathology. J. Oper. Res. Soc. **35**(1), 7–25 (1984)
21. Zimmermann, E., Bril El Haouzi, H., Thomas, P., Thomas, A., Noyel, M.: A batching and scheduling optimisation for a cutting work-center: Acta-Mobilier case study. IESM (2017)

Part II
Big Data Management

Communicating Aircraft Structure for Solving Black-Box Loss on Ocean Crash

Kais Mekki, William Derigent, Eric Rondeau and André Thomas

Abstract Commercial aircrafts use black-box required for crash investigation purposes. While a black-box can be easily recovered in crash events on land, the same does not apply to crash events in great deep ocean water. This paper presents a new solution towards solving black-box data loss on ocean crash using a paradigm called communicating materials. The solution is developed through uniformly integrating hundreds of micro sensors nodes in the aircraft structure. The nodes could then construct a Wireless Sensor Network (WSN) inside the aircraft. When a crash is detected by the aircraft system, the black-box data could be stored in all nodes using data storage protocols for WSN. Since nodes are uniformly deployed in the whole aircraft structure, investigators could thus gather preliminary crash causes information from the nodes inside any floated aircraft wreckage in the ocean. This solution was evaluated using Castalia simulator in terms of reliability, storage capacity, and energy efficiency.

Keywords Aircraft black-box · Wireless sensors networks · Storage protocols Clustering · Systematic-Reed-Solomon

K. Mekki (✉) · W. Derigent · E. Rondeau · A. Thomas
Research Centre for Automatic Control of Nancy, CNRS UMR 7039,
Campus Sciences, BP 70239, 54506 Vandoeuvre-lès-Nancy Cedex, France
e-mail: kais.mekki@univ-lorraine.fr

W. Derigent
e-mail: william.derigent@univ-lorraine.fr

E. Rondeau
e-mail: eric.rondeau@univ-lorraine.fr

A. Thomas
e-mail: andre.thomas@univ-lorraine.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_7

1 Introduction

According to international rules, commercial aircrafts must be provided with a unit generally known as a “black-box” having means for recording data regarding the aircraft for crash investigation purposes [1]. While the black-box can be easily recovered in crash events on land, the same does not apply to crash events in the ocean where the problem is recovering this object in great deep waters. As example, the Air France flight 447 crashed in the Atlantic Ocean on 1 June 2009, while flying from Rio de Janeiro in Brazil to Paris in France. In fact, the black-box is fitted with an Underwater Locator Beacon (ULB) that begins to radiate an acoustic signal at 37.5 kHz if its sensor touches water. They work to a depth of just over four kilometres, and can ping once a second for 30 days before the battery runs out. During 30 days after Air France flight 447 crashed, a French submarine was used to listen to the acoustic signal emitted by the ULB in a search area centred on the aircraft’s last known position, without any success. However, it took search teams two years (until 2 May 2011) to find and raise the black-box at a cost of 40 million dollars [2]. Another case is the disappearing of Antonov An-72 flight on 22 December 1997, while flying from Abidjan in Côte Ivoire to Rundu in Namibia. The airplane vanished over the South Atlantic Ocean, but since the black-box has not been found, the reason of the disappearance remains unknown. One notable event is Malaysia Airlines Flight 370 that disappeared in Indian Ocean on 8 March 2014, while flying from Kuala Lumpur, Malaysia to Beijing, China. The flight black-box was found, so the incident’s cause remains undetermined [3].

Considering the advances of modern communication technology, researchers and airframe manufacturers recommended thus for flights that black-boxes should be extended by enabling data transmission from the black-box to a Cloud through satellite communication [4], by prolonging the ULB battery lifetime from 30 to 90 days and by increasing the beacon radio range [3], or by ejecting the black-box prior to the aircraft crash and providing it with specific water floating tools [1]. However, all these proposals have not been implemented nowadays in commercial aircrafts.

This paper is oriented towards solving said drawbacks using a paradigm called “communicating material” invented in the Research Centre for Automatic Control of Nancy (University of Lorraine, France) since 2009. The communicating material enhances a classic product material with the following capabilities: it can store data, communicate information at any point of its surface, and keep these previous properties after physical modifications. Indeed, the product does not communicate using some electronic devices in specific points, but becomes intrinsically and continuously communicating. To meet this vision, thousands of micro/nano electronic devices (e.g. RFID, WSN) are inserted into the material of the product during its manufacturing. Projects were developed using this concept including communicating concrete [5], communicating textile [6], wood traceability [7], and self-measurement system [8].

In this paper, we define new application of communicating material paradigm for aircraft systems which could store the black-box data into its structure using WSN. In the literature, WSN are used mainly for aircraft structural health monitoring, aircraft hydraulic monitoring systems, and aircraft engine health management [9]. In this paper, we propose also a new application of WSN in aircraft systems.

This application is developed through uniformly integrating a multitude of micro sensors nodes in the aircraft structure as shown in Fig. 1. When a crash is detected, the black-box data are disseminated and then stored in the integrated WSN. The application guarantees that black-box data are present in each sensors node. So, information could be read in all pieces of the aircraft structure since the nodes are uniformly deployed. Thus, investigators could gather preliminary crash causes information from the nodes inside any floated aircraft wreckage in the ocean.

To disseminate/store the black-box data in all nodes of the aircraft structure, a cluster-based protocol in WSN is proposed in this paper. In this protocol, we employ the technique of powerful sensors nodes (cluster-heads) in conjunction with ordinary sensors nodes (cluster-members) to perform distributed data storage. Moreover, to improve the dissemination reliability in WSN, the Systematic-Reed-Solomon code is used. The proposed solution is evaluated using the Castalia simulator by studying the data reception reliability, the storage capacity, and the energy consumption.

The rest of paper is organized as follows. Section 2 explains the problem statement. Section 3 presents a state-of-the-art of existing cluster-based storage techniques in WSN. Section 4 details the architecture and the communication model of the proposed solution. Section 5 presents the simulation results. Finally, Sect. 6 discusses and concludes the paper.

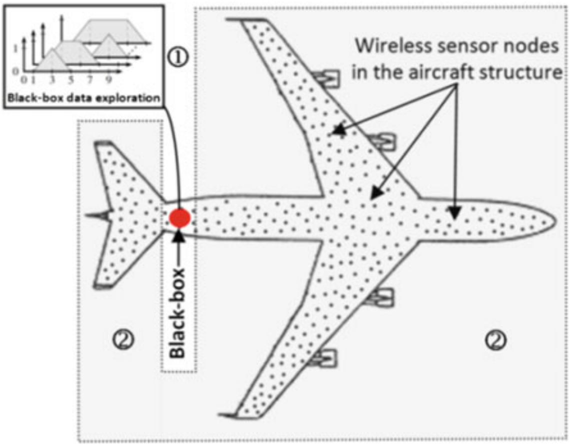


Fig. 1 Uniform deployment of wireless sensors nodes in the aircraft structure

2 Problem Statement

Currently, commercial aircrafts are equipped with two black-box recorders. One of these, the Cockpit Voice Recorder (CVR), records the last 2 h of radio transmissions and sounds in the cockpit, such as the pilot's voices and engine noises [10]. The other, the Flight Data Recorder (FDR), retains 25 h of mandatory flight parameters such as engine parameters and cockpit controls data [11].

The CVR allows 2 h of audio recording which leads to big data size, and it would be difficult to store it in wireless sensors node memory chip. Thus, the study of this paper is limited to FDR data storage which brings to investigators preliminary idea of the cause of the aircraft crash. In the rest of this paper, the "black-box" word means the FDR recorder.

Indeed, FDR contains too many parameters which could be also difficult to store all of them in a sensors node memory chip. Thus, a solution is proposed by extracting the relevant parameters data from the FDR. Only the most relevant data are stored throughout the WSN in the aircraft structure. In Fig. 1, two areas are presented:

① **Black-box data exploration**: The black box is explored to select the relevant data parameters that must be stored in the aircraft structure. The relevant data presents the most representative parameters that allow identification of aircraft crash causes. For this issue, we propose to uses a previous algorithm developed in the Research Centre of Automatic Control of Nancy [6]. In [6], the authors propose a data dissemination process to select context-sensitive information from a database that must be stored/replicated on intelligent product constructed using communicating materials paradigm. The approach uses the fuzzy-AHP theory for aggregating points of view from different actors. In the context of this paper, actors could be experts of aircraft crash investigations. The approach consists in assessing the relevance of storing a given data on the product. Higher relevance value data have the priority to be stored first in the product.

② **Data storage in WSN**: In this phase, the selected relevant data are disseminated to all nodes in the aircraft structure through WSN communication protocol. Upon receiving the data, each node stores them in its memory chip.

In the scientific literature and industrial field, no information is published about the data structure inside the black-box for security purposes. Thus, *the black-box data exploration phase is not studied in this paper*. However, this paper presents a communication protocol solution for data storage phase.

Indeed, the nodes inside the aircraft are used to store black-box data when a crash is detected. The deployed sensors nodes could be useful for other purposes such as aircraft Structural Health Monitoring (SHM) of the aircraft structure during all its lifecycle. The nodes could sense and monitor the internal parameters like cracks, strain, torque and displacement as well as external parameters like temperature, pressure, magnetic field, etc.

The key focus of our study is to ensure the existence of black-box data in each piece of the aircraft wreckage on ocean crash. Therefore, the black-box data should be efficiently and uniformly stored into all nodes of the aircraft structure (i.e. data should be present in each piece of the aircraft structure).

We define also the *warning time* which is the time elapsed from the instant of the crash detection, to the impact of the aircraft with the ground/ocean. In [3], the statistic results indicate that the warning time is greater than 15 s in 75% of the crash cases, greater than 30 s in 59% of the cases, greater than 60 s in 34% of the cases and greater than 120 s in 23% of the cases. Thus, the black-box data storage process in WSN has to ensure very low delay. Hence, WSN could store as much as possible of the relevant black-box data during the warning time.

The aim of this paper is to store the black-box data in all nodes, in order to make data present in each piece of the aircraft structure. As a result, the next section of the paper focuses on data storage techniques in WSN.

3 Data Storage Techniques in WSN

Considerable research has been conducted on data storage for WSN. To store data throughout the entire WSN, different techniques are proposed in the literature [5]: Flooding-based, Random-Walk-based, Tree-based, and Cluster-based.

Authors in [12] and [13] evaluated these techniques and showed that cluster-based approach can highly decrease the storage process delay, the storage uniformity throughout the WSN, and the overall system scalability.

Clustering techniques for data storage are widely studied in the literature. In [14], authors divide the entire WSN field horizontally and vertically into small rectangular zones. Each zone has a cluster-head which functions as the server for all other nodes in the zone. The head maintains the links to the nodes within its zone, and the heads in its neighbour's zones. In order to save energy, authors assume every node has two interfaces to adapt to different transmission range: a *short range interface* and a *long-range interface*. The short-range interface is used most of the time and enables nodes to communicate with neighbours. The long-range interface enables adjacent head nodes to directly route messages when necessary. After sensing an event, the node sends the data to cluster-head which then forwards and stores it in some cluster-heads of other zones using the long-range interface.

In [15], authors organize the nodes into different clusters and determine the data storing node in each cluster (i.e. each cluster has a representative node as a data storage reference). Given N sensors nodes, authors applied *k-means* clustering method to divide N into k clusters, and determine k data storage nodes which have minimum accessing cost. When all nodes and clusters association are established, data replication in storage nodes could be done through communication between cluster-heads.

The above approaches can cause the cluster-heads to overload with the increase in nodes density. The cluster-heads close to the source or the storage nodes are more likely to expend their battery earlier than other nodes due to the intersection of multi-hop routes and concentration of messages traffic in these cluster-heads. In WSN, this problem is referred to as the hotspot problem. In the literature, authors are aware of this problem. Thus, they change periodically the cluster-heads in order to balance the storage load between all the nodes [16]. However, periodically changing the cluster-heads incurs supplementary load cost which arise more energy consumption and decrease of WSN lifetime.

Thus, another solution was proposed by employing powerful nodes to mitigate hotspot problem. As an example, authors in [17] employ powerful nodes to perform distributed data storage in WSN. All sensors are uniquely identified and can be of two types.

- The first one, named *L-sensor* for low-end sensor, is a node with limited resources including processor, storage, and communication power.
- The second one, named *H-sensor* for high-end sensor, is a node with much higher resources. Authors take advantage of the powerful communication capability of H-sensors to improve data distribution/storage. H-sensors are responsible for forwarding the data messages from the source toward the L-sensors.

There are a few other studies with the same aforementioned focus that use more powerful nodes in conjunction with ordinary sensor nodes to perform distributed data storage [18].

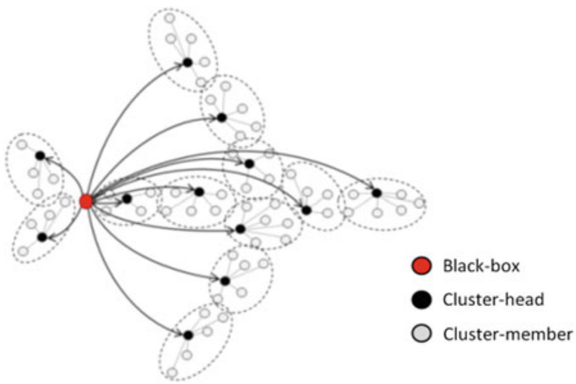
In this paper, a cluster-based communication model is used to propose a solution for relevant black-box data storage in WSN. We employ the technique of powerful sensors nodes (cluster-heads) in conjunction with ordinary sensors nodes (cluster-members) to perform distributed data storage. Moreover, we employ Systematic-Reed-Solomon coding technique to improve the data dissemination reliability.

The remainder of the paper describes the proposed cluster-based solution in detail, and then, presents the simulation results obtained for data reception reliability, storage capacity, and energy consumption.

4 Cluster-Based Solution for Black-Box Data Storage

In this section, the nodes characteristics and the messages exchange scheme are firstly presented. Then, the proposed data survivability scheme is detailed using Systematic-Reed-Solomon code.

Fig. 2 Cluster-based communication model for black-box data storage



4.1 Nodes

To organise the network in different clusters, two types of sensors nodes are employed as shown in Fig. 2.

The first one, named *CM* for *cluster-member*, is a node with normal resources, including processor, storage, limited communication range, and energy resources. The second type, named *CH* for *cluster-head*, is a node with more sophisticated resources. *CH* nodes have improved processing, storage, battery and communication power when compared with *CM* nodes.

Hence, it is assumed the network is composed of n_{CM} *CM* nodes, and n_{CH} *CH* nodes, where $n = n_{CM} + n_{CH}$ and $n_{CM} \gg n_{CH}$. All *CM* and *CH* nodes uniformly deployed throughout the aircraft structure as shown in Fig. 2. Each *CH* covers a subset of *CM* nodes. Thus, each *CM* node is associated to its closest *CH*. Each *CM* node i communicates only with its associated *CH* node j that is inside its communication radius r_1. The distance between i and j should be less than or equal to r_1 ($d(i, j) \leq r_1$). *CH* nodes are equipped with two radios, each one with different frequency and a different communication radius (r_1 and r_2 where $r_2 \gg r_1$). It is also assumed that radio frequencies do not interfere with each other. The *CH* node can communicate with both black-box and its associated *CM* nodes inside communication radius r_2 and r_1, respectively.

4.2 Messages Exchange Scheme

Figure 3 shows the communication between the black-box, the *CH*, and the *CM* nodes for data storage from the crash detection instant until the crash of the aircraft in the ground/ocean.

Three messages are used:

- *Crash_Start_MSG*: It is used to notify the crash detection to all nodes.

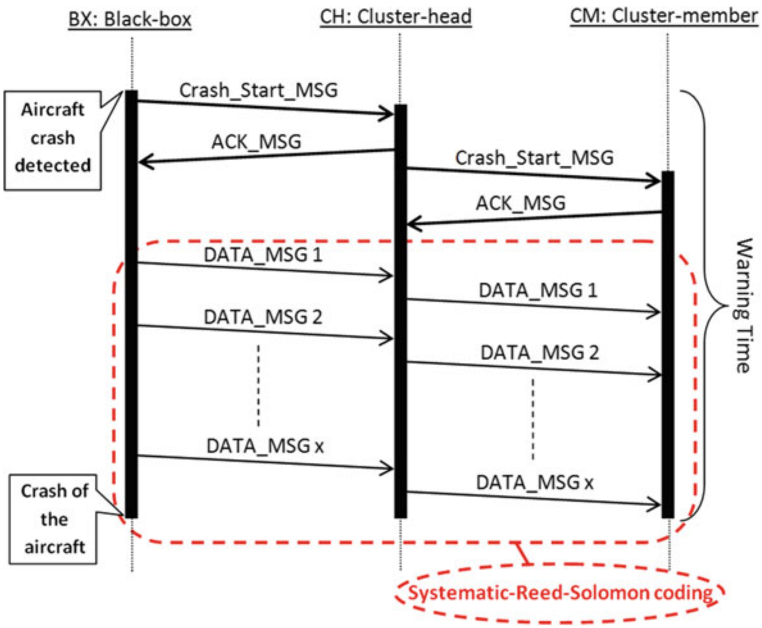


Fig. 3 Message exchange between nodes for black-box data storage

- *ACK_MSG*: It is used to acknowledge the reception of *Crash_Start_MSG*.
- *DATA_MSG*: It is used to carry the black-box data.

When the crash is detected by the aircraft system, a *Crash_Start_MSG* message is broadcasted by the black-box toward the *CH* nodes. Upon receiving this message, each *CH* sends an *ACK_MSG* message to black-box for acknowledgment. Then, each *CH* broadcasts the *Crash_Start_MSG* message toward its associated *CM* nodes. Upon receiving this message, each *CM* sends *ACK_MSG* to *CH* for acknowledgment. Thus, all *CH* and *CM* nodes are awarded of the crash and still in continuous active mode and wait for reception of data messages.

When *Crash_Start_MSG* message dissemination is finished, the black-box starts successive broadcast of data messages *DATA_MSG* to *CH* nodes which broadcast them to its *CM* nodes. Thus, each *CH* is a relay node between the black-box and its *CM* nodes. Upon receiving a *DATA_MSG*, each *CM* stores the carried data and then drops the message.

As shown in Fig. 3, Systematic-Reed-Solomon coding technique is used for black-box data to increase *DATA_MSG* reception reliability. In the following, the Systematic-Reed-Solomon code is detailed for black-box data storage.

4.3 Systematic-Reed-Solomon for Black-Box Data Storage

Systematic-Reed-Solomon code (SRS) [19] is an error control scheme for handling messages losses in real-time communication of WSN. In SRS, the sender divides the data into m fragments which are encoded by adding another k fragments. These $m + k$ fragments, called *code words*, are transmitted to the receiver which is able to reconstruct the original data out of them even if some fragments were lost. The receiver is able to reconstruct the original data only if the number of fragments it has received is equal to or greater than the number of original fragments (i.e. m). Thus, SRS can achieve high reliability without retransmission.

SRS can be represented as multiplication of a matrix and a vector, as shown in Fig. 4. Here the matrix A is a coding matrix (where each x_i is nonzero and distinct from each other), W is a vector of data fragments, and code words are contained in a vector Z. If we have any m rows of A and their corresponding Z values, we can obtain the vector W which contains coefficients of the polynomial, which is again the original data. To encode/decode in SRS, the addition is simply the XOR of two numbers. The multiplication and division operations are done under the arithmetic of the Galois Field (called also Finite Field) [20].

In this paper, SRS code is used as storage survivability technique in sensors nodes at the cost of increased memory storage resource. The relevant black-box data are fragmented, and each fragment is then encoded producing m original data code words and k additional redundant code words. All these code words are sent separately in *DATA_MSG* messages to be stored in *CM* nodes of the aircraft structure. The carried code words are stored only if the *DATA_MSG* is correctly received. The decoding process can be done by aircraft crash investigators which extract stored code words from survived nodes and decode them to reconstruct the original black-box data. Thus, no coding/decoding computational effort is needed for all sensors nodes. Storage of encoded code words increase the black-box data survivability, since missing blocks of original data bits in a node could be recovered through decoding operation or it could be found in other survived nodes.

Indeed, if one *DATA_MSG* carries one code word, each code word should be very large. This makes the implementation intractable since SRS operations on such a large field require huge space and time in black-box which could delay the data

$$\underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \\ 1 & x_{m+1} & \cdots & x_{m+1}^{m-1} \\ 1 & x_{m+2} & \cdots & x_{m+2}^{m-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^{m-1} \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{pmatrix}}_{W} = \underbrace{\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \\ p(x_{m+1}) \\ p(x_{m+2}) \\ \vdots \\ p(x_n) \end{pmatrix}}_{Z} \begin{matrix} \left.\vphantom{\begin{matrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{matrix}}\right\} m \\ \left.\vphantom{\begin{matrix} p \\ p \\ \vdots \\ p \end{matrix}}\right\} k \end{matrix}$$

Fig. 4 Systematic-Reed-Solomon coding

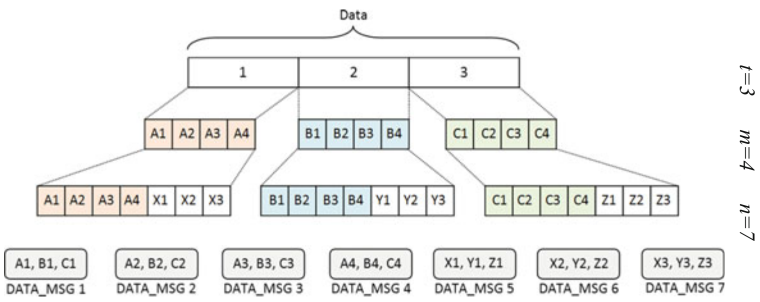


Fig. 5 Dividing block of black-box data into multiple independent code words, and their handling by multiple DATA_MSG messages

dissemination start when an aircraft crash is detected. The solution would be to use small code words size. Then, however, the payload in the *DATA_MSG* message gets too small. By putting multiple independent code words into a *DATA_MSG* message, we can fully utilize payload space of *DATA_MSG* without problems of large code word. Imagine dividing one big data into t small pieces of data. Then each data is again divided into m fragments, and encoded into n code words. We have total of $t \times n$ code words to send. Pack the ith code word from each independent t fragments into a single *DATA_MSG* message. Any n *DATA_MSG* messages will provide n code words for all t fragments, and original t data could be reconstructed. Figure 5 shows an example for $t = 3$, $m = 4$, and $n = 7$.

5 Simulation Results

In this section, we discuss the performance of the proposed solution through computer simulation. The simulation settings are firstly detailed, and then the simulation results are presented and discussed.

5.1 Simulation Setup

The proposed solution was implemented using Castalia. In this simulator, we modelled the MicroStrain node which is presented as an ideal sensors node for aircraft SHM applications [9]. Wireless radio channel characteristics such as signal noise, interference ratio, and average path loss are chosen to simulate the realistic modelled radio wireless channel in Castalia. The simulated aircraft structure is Airbus A320.

In this simulation, fragments of black-box data are 48 bytes long. Code words are divided to 8 bit-long units, and thus there are 48 original code words and 48 additional redundant codes (i.e. for high reliability level, the number of redundant codes should be equal to or greater than the number of original codes as discussed in [19]). Therefore, each *DATA_MSG* message contains 96 bytes; each byte is a code word from independent fragments as discussed in Fig. 5.

The proposed cluster-based protocol operates under a heterogeneous WSN made up of two types of sensor nodes, *CH* and *CM*. *CH* nodes have an increased communication capacity when compared with *CM* nodes. Therefore, we modulate firstly a *CH* node that uses two radios whose frequencies do not interfere with each other.

We uses 2.405–2.470 GHz frequency band spread spectrum over 14 channels. The first channel is reserved for the communication between *CH* nodes and black-box with long transmission range. The rest of channels are used for the communication between *CH* and *CM* with short transmission range. To communicate with *CM*, the adjacent *CH* nodes do not uses the same channel to avoid interference between clusters.

The *CH* and *CM* nodes are uniformly deployed in the simulated aircraft structure. *CH* in each 9 m^2 is simulated. Various densities of *CM* nodes are simulated: 1 node each 1 m^2, 2 nodes each 1 m^2,…, 12 nodes each 1 m^2. The simulated warning times are 15, 30, 60, and 120 s. In the following, the proposed solution is evaluated in terms of data reception reliability, storage capacity, and energy consumption.

5.2 Data Reception Reliability

To assess the data reception reliability, we disseminated the code words of a data block using the cluster-based protocol. Simulation results show that each node has a medium probability of 0.65 to receive each disseminated message. Moreover, the nodes densities do not have noticeable impact on the reception probability.

To reconstruct each data block during aircraft crash investigation using SRS, the number s of stored code words in CM should be equal to or greater than the number m of original codes ($s \geq m$). Figure 6 shows the average number of stored codes words in CM nodes. It shows also the average number of included original codes. For all nodes densities, Fig. 6 shows that the number of stored code words is around 60. This number is higher than the number of sent original code words $m = 48$. Thus, the original black-box data block could be successfully decoded using the stored code words in any sensors node.

In conclusion, the medium reception probability problem is bounced by employing the SRS code, which ensures the recovery of black-box data from any node even if subset of DATA_MSG messages were lost during wireless transmission.

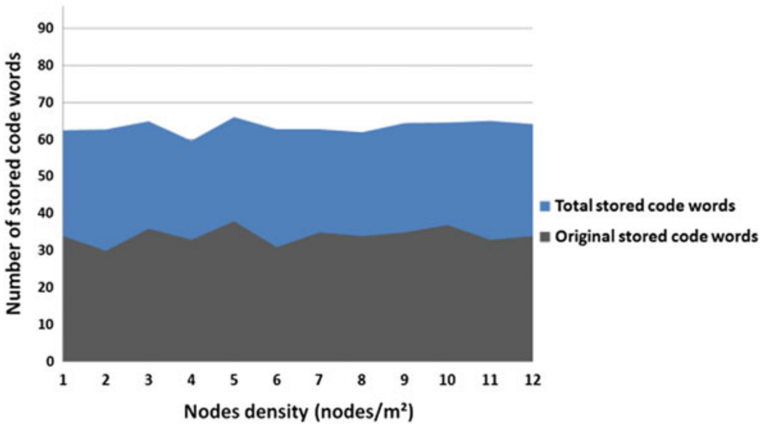


Fig. 6 Amount of stored code words in *CM* for an encoded data block

5.3 *Storage Capacity*

The storage capacity metric provides an idea of how much of data can be stored by the proposed solution during the warning times. Figure 7 shows the average amount of stored data in each node (solid lines) for the different nodes densities during the warning times. It shows also the average amount of decoded original data (dotted lines) among the stored data in each *CM* for the simulated warning times.

Figure 7 shows that the nodes' density does not impact the amount of stored bytes. The protocol stores in each node 1.151 Mbytes during 120 s, 575.67 Kbytes

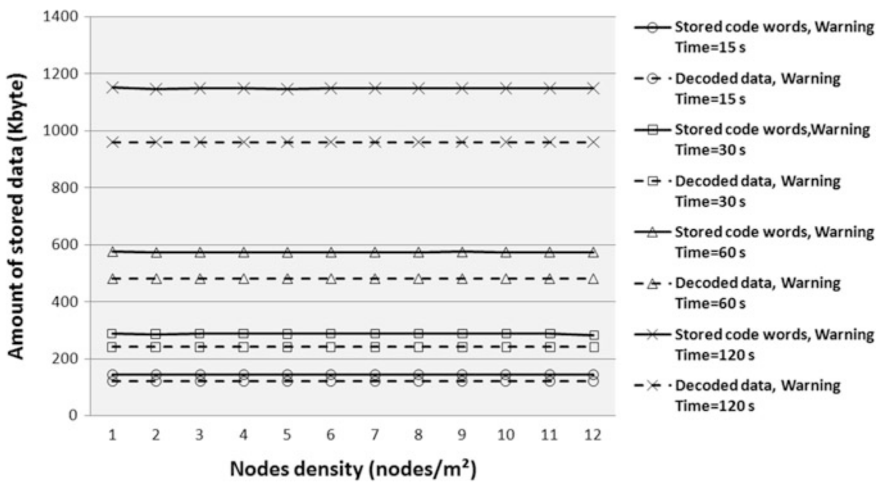


Fig. 7 Amount of stored data in each *CM* during the warning times

during 60 s, 287.67 Kbytes during 30 s, and 143.72 Kbytes during 15 s. The amount of decoded original data is lower than the total stored bytes since the nodes store bytes including both original code words and redundant code words. After decoding, each node carries 959.72 Kbytes during 120 s, 479.72 Kbytes during 60 s, 239.76 Kbytes during 30 s, and 119.76 Kbytes during 15 s.

In fact, commercial aircraft models use FDR systems which store 64 words per second of 12 bits each over a 25 h period [11]. Thus, the proposed cluster-based solution allows storing data that are recorded during 20.79 min in the FDR for the warning times 15 s. It allows storing data recorded during 41.62 min in FDR for the warning times 30 s, data recorded during 1.38 h in FDR for the warning times 60 s, and data recorded during 2.77 h in FDR for the warning times 120 s.

5.4 Energy Consumption

The energy consumption metric provides an idea of how much energy should be left in sensors nodes when aircraft crash is detected. Simulation results show that the consumed energy increases with the warning time. However, for each warning time period, the consumed energy is the same for all nodes since they are in continuous active mode. All the nodes have their radios active during the warning time (i.e. nodes do not sleep). So either receiving, idle listening, or transmitting, the nodes consume the same amount of energy.

The consumed energy is equal to 1.019 J for the warning time 15 s, equal to 2.039 J for 30 s, equal to 4.079 J for 60 s, and equal to 8.159 J for 120 s. These low energy amounts should be left in each sensor node battery when aircraft crash is detected to ensure data dissemination and storage in WSN of the aircraft structure.

6 Conclusion

This paper is oriented towards solving the problem of black-box recovery in deep ocean water during aircraft crash investigation, using communicating material paradigm. The solution consists in uniformly integrating hundreds/thousands of micro sensors nodes in the aircraft structure. In crash detection, the black-box data could be replicated throughout all nodes inside the aircraft structure using storage protocol for wireless sensors networks. The proposed storage protocol uses cluster-based communication model and employs the technique of powerful sensors nodes in conjunction with ordinary sensors nodes to perform distributed data storage. To improve reliability, the Systematic-Reed-Solomon code is used. The overall idea was evaluated using Castalia simulator.

However, our future work focuses on testing this solution in Raspberry PI platform nodes.

References

1. Santolalla, E.V., Cortes, D.F., Gonzalez, E.M., Fernandez, E.G., Abaurrea, P.G.: Aircraft black box, US patent US8489259B26 (2013)
2. French BEA: Final report on the accident on 1st june 2009 to the airbus A330–203 registered F-GZCP operated by air France flight AF 447 Rio de Janeiro—Paris (2012)
3. French BEA: Triggered transmission of flight data working group, Technical Report (2011)
4. Saravanan, S., Ragulkannan, R., Sreenivasaraja, N.: Intelligent method of flight data recording using satellite communication by avoiding the search of black box. Int. J. Curr. Res. Mod. Educ. **1**(2), 238–241 (2016)
5. Mekki, K., Derigent, W., Zouinhki, A., Rondeau, E., Thomas, A., Abdelkrim, M.N.: USEE: a uniform data dissemination and energy efficient protocol for communicating materials. Future Gener. Comput. Syst. **56**, 651–663 (2016)
6. Kubler, S., Derigent, W., Thomas, A., Rondeau, E.: Method for embedding context-sensitive information on communicating textiles via fuzzy AHP. Intell. Fuzzy Syst. **26**, 597–610 (2014)
7. Jover, J., Thomas, A., Leban, J.M., Canet, D.: Interest of new communicating material paradigm: an attempt in wood industry. J. Phys. **416**(1), 12–31 (2013)
8. Peña, R., Krommenacker, N., Charpentier, P.: A new strategy for dimensional metrology using smart dust networks. In: International Conference on Indoor Positioning and Indoor Navigation (2011)
9. Vujić, D.: Wireless sensor networks applications in aircraft structural health monitoring. Appl. Eng. **13**, 79–86 (2015)
10. American Federal Aviation Administration (FAA): Revisions to cockpit voice recorder and digital flight data recorder regulations, Technical Report (2005)
11. French BEA: Flight data recorder read-out: technical and regulatory aspects, Study Report (2005)
12. Tunca, C., Isik, S., Donmez, M., Ersoy, C.: Distributed mobile sink routing for wireless sensor networks: a survey. IEEE Surv. Tutorials **16**(2), 887–897 (2014)
13. Mekki, K., Derigent, W., Zouinkhi, A., Rondeau, E., Thomas, A., Abdelkrim, M.N.: Non-localized and localized data storage in large-scale communicating materials: probabilistic and hop-counter approaches. Comput. Stand. Interfaces **44**, 243–257 (2016)
14. Wang, J.-H., Yu, Y., Naemeka, M.C.: Data storage algorithms based on clustering in wireless sensor networks. J. Netw. **8**(8), 1796–1802 (2013)
15. Jaichandran, R., Irudhayaraj, A., Raja, J.: Effective strategies and optimal solutions for hotspot problem in wireless sensor networks. In: 10th IEEE International Conference on Information Sciences Signal Processing and their Applications, pp. 389–392 (2010)
16. Xin, Y., Zhang, R.: An energy-efficient mobile sink routing algorithm for wireless sensor networks. In: 7th IEEE Conference on Wireless Communications and Mobile Computing, pp. 1–4 (2011)
17. Maria, G., Guidoni, D., Viana, A.C.: A distributed data storage protocol for heterogeneous wireless sensor networks with mobile sinks. Ad Hoc Netw. **11**, 1588–1602 (2013)
18. Sheng, B., Li, Q., Mao, W.: Data storage placement in sensor networks. In: 7th ACM Symposium on Mobile Ad Hoc Networking and Computing, pp. 344–355 (2006)
19. Kim, S., Fonseca, R., Culler, D.: Reliable transfer on wireless sensor networks. In: 1st IEEE Conference on Sensor and Ad Hoc Communications and Networks, pp. 449–459 (2004)
20. Awami, L., Hassanein, H.: Energy efficient data survivability for WSNs via decentralized erasure codes. In: 37th IEEE Conference on Local Computer Networks, pp. 577–584 (2012)

Data Management Architectures for the Improvement of the Availability and Maintainability of a Fleet of Complex Transportation Systems: A State-of-the-Art Review

Damien Trentesaux and Guillaume Branger

Abstract This paper deals with the way constructors of a fleet of complex systems, and especially, in the field of transportation, aim to improve the availability and the maintainability of their fleet during their use, fleet operated by a public operator or a private company. More precisely, the focus is set on the architecture of the data management system that supports this aim. In this context, this paper presents a literature review on the main existing approaches that address the architecting of the data management of fleet or a set of transportation systems with the target to improve directly or indirectly their availability or maintainability. For that purpose, a positioning typology is suggested, which enables to identify, characterize and evaluate the state-of-the-art from the point of view of an industrialist working in the transportation sector. A set of future research challenges are finally suggested.

Keywords Data architecture · Data management · Architecture Transportation · Availability · Maintainability · Review

1 Introduction

This paper deals with the way constructors of a fleet of complex systems, and especially in the field of transportation (road, rail, aircraft and naval sectors), can improve the availability and the maintainability of their fleet during their use, the fleet being operated by a public operator or a private company. In transportation, the

D. Trentesaux (✉)
LAMIH UMR CNRS 8201, SurferLab, University of Valenciennes, Le Mont Houy, 59313 Valenciennes Cedex, France
e-mail: damien.trentesaux@univ-valenciennes.fr

G. Branger
Bombardier Transport, 1 Place Des Ateliers, 59154 Crespin, France
e-mail: guillaume.branger@rail.bombardier.com

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_8

availability (the proportion of time a system is in functioning condition) and the maintainability (the ability to maintain a system) are high-stakes global indicators. Addressing availability and maintainability at fleet level is nowadays among the most challenging activities identified by researchers [1].

In this context, one critical aspect concerns the development and the integration into modern data management systems of algorithms and methods dealing with the monitoring, diagnosis, prognosis, health-status assessment of equipment coupled with reparation and overhaul activities led by maintenance centers and organizations. Data management systems and more globally, enterprise data management systems, have recently regained interest with the emergence of Big Data issues [2]. In our work, a *data management system* is an information system aiming to gather, memorize, manipulate and communicate data, information and digitalized knowledge coming from physical equipment and systems to users, managers or other information systems of a company. As stated, we deal more specifically with data management systems devoted to the monitoring, evaluation, control and improvement of the availability and the maintainability of a fleet. Designing such a data management system is a delicate task and early design choices are key elements to get a useful data management system.

From our perspective, one key point is the design of the architecture of such data management systems. We consider that a *data management architecture* describes the functions and algorithms (software) to be supported as well as the localization of the different computing elements (hardware) in charge of the flow of data, information and digitalized knowledge of a data management system. The data management architecture deals then with the global organization of the data/information/knowledge flow and its integration with other existing information systems. It answers the question: "which computing device will do what, where, when and with whom?"

The aim of the authors is to specify and develop an original data management architecture of a fleet of complex transportation systems dealing with at least the two introduced global indicators (availability and maintainability). Given the high stakes in transportation, and especially the ones relevant to big data management and cyber-physical systems design approach, the authors consider of major importance the definition of a correct picture and the accurate evaluation of existing main architecture design approaches from the industrial partner's point of view (train transportation sector) before specifying theirs. Defining such a correct picture along with an accurate evaluation is the topic of this chapter.

This chapter presents thus a literature review on the main design approaches that address more or less explicitly the architecture of the data management of a fleet or a set of transportation systems with the target to improve directly or indirectly their availability or their maintainability. The criteria for selecting papers have thus been the following (AND aggregator):

- A selected paper must deal with a single transportation system or preferably, a fleet of. These systems must be in use, preferably in the transportation sector, but not in a compulsory way to get the best practices from other historical application fields.
- It must deal with objectives relevant to the estimation and the improvement of the availability or the maintainability of the systems.
- It must address data management issues, from sensors to high-level decisions with a focus on diagnosis, prognostics and health management (PHM), remaining useful life (RUL) estimation, maintenance decision, etc.
- Lastly, it must contain sound elements describing the architecture of the suggested data management system.

There is no criterion related to the modelling approach (e.g., data driven or model driven), neither to the technological solutions used to deploy the architecture (e.g., cloud technology, web-services, cyber-physical systems, etc.).

In this context, we realized that the generic approach suggested in [3] aiming to position different decisional architectures, could be used as a typology to position the corresponding reviewed state-of-the-art. Specifically, we identified three types of architecture design in our literature review, which we named in the remaining of this paper “the pure centralized approach”, “the mixed centralized/flat hierarchical approach”, and “the semi-heterarchical approach”.

The following part describes thus our review based on this typology. Each part is organised as follows: first, a definition of the type of architecture is proposed; then a study of some representative contributions is realised, followed by a synthesis of the strengths and weaknesses of the type of architecture, elaborated from the point of view of train transportation industry. From this review, we identified some future research challenges, which and are presented in the last part of this paper.

2 The Pure Centralized Approach

2.1 Definition

When the architecture is designed according to this approach depicted in Fig. 1, a raw data collection, coming from an equipment or systems composing the fleet, occurs in a centralized data base (using for example cloud technologies) in which remote data treatments are realized to optimize the availability and the maintainability of these equipment or systems. The data management architecture is then fully centralized.

A growing number of articles address this kind of architecture. We detail hereinafter some of representative contributions articulated according to the industrial sector in which they are applied.

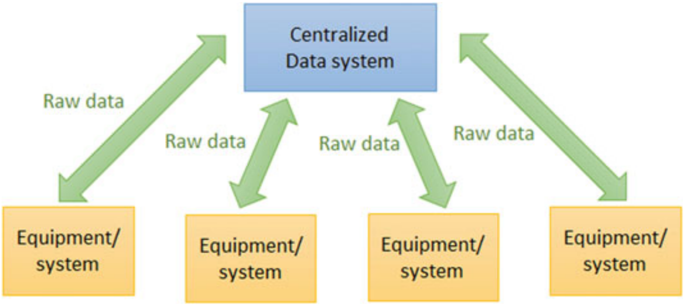


Fig. 1 The pure centralized approach for developing data management

2.2 Contributions

Aviation/aircraft sector: In [4], the authors state that centralized big data management for the health management in the aircraft sector is promising but still in its infancy stage. They suggest the design of a centralized big data analysis and application platform for civil aircraft health management. Two other similar approaches for PHM of aircrafts are proposed by Yang et al. [5] and Larsen [6].

Rail sector: In their interesting paper, [7] address the issue of track defect detection. They suggest using the axle box acceleration system of a train, but such a system can result in over 100 TB of raw data per day. To handle this, they use different techniques, including data reduction ones. Even with these technics, parallel computing is required to be sufficiently responsive. In [8], the authors deal with the problem of condition-based maintenance applied to the condition monitoring and predictive maintenance of train axle bearings based on sensors data collection, with the purpose of maximizing their RUL using a big-data centralized approach. They develop an online Support Vector Regression for prediction of the RUL. In [9], the authors addressed the health estimation of loco mechanical transmission components based on the acquisition and centralized post-processing of the stator currents of an induction traction motor monitored by standard on-board automation systems. They used a statistical method named Independent Component Analysis (ICA) for decomposing an observed complex dataset into components as much as possible statistically independent from each other. From our perspective, a flagship illustration of a pure centralized approach in the rail sector is the Alstom Traintracer [10], see Fig. 2. This figure clearly highlights the centralization of raw data for remote data analytics. An interesting, noticeable aspect is that Traintracer addresses the fleet level, rarely addressed in the literature.

Another interesting centralized approach is suggested in the paper proposed by Thaduri [11]. The authors provide an original analysis of the use of big data in railway assets based on the estimation of RUL from different sources. From their review, they suggest to adopt a multi-modelling vision when handling in a

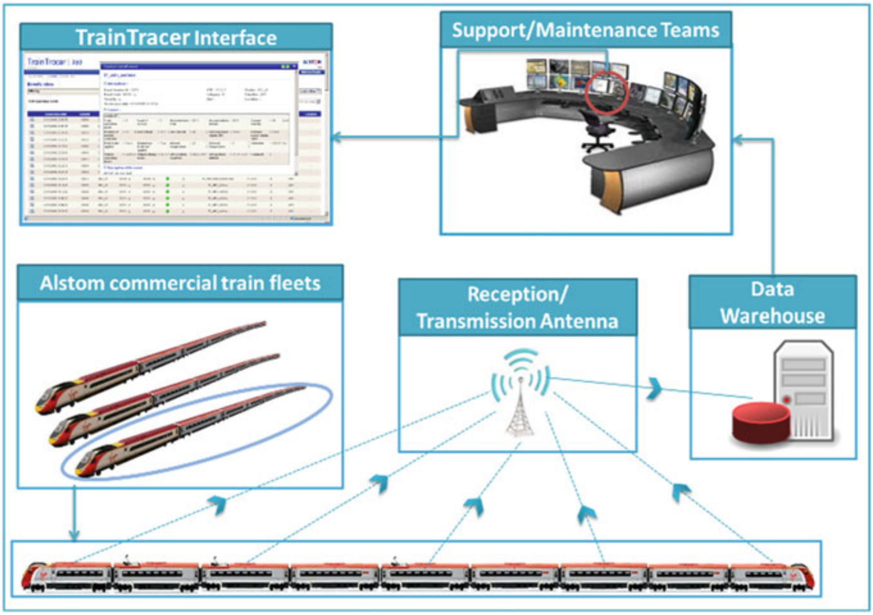


Fig. 2 A flagship illustration of the centralized data management services in train transportation, the Alstom Traintracer

centralized architecture big data in railway management (physics based, symbolic based and data-driven based methods), as depicted in Fig. 3.

Automotive sector: In [12], the authors designed a centralized proactive real-time traffic monitoring strategy evaluating operation and safety simultaneously of urban expressways aiming to reduce congestion and crash risk. They used big data analytics from Microwave Vehicle Detection System (MVDS) deployed on an expressway network in Orlando.

Shipbuilding/maritime sector: In [13], the authors propose a framework integrating an Internet-of-Things (IoT) approach with computing technologies to support remote centralized big data analytics for the maritime industry.

Manufacturing and logistics sectors: Yuanyuan and Jiang [14] propose an equipment predictive maintenance model based on big data analytics in a pure centralized architecture. A general regression neural network evaluates the relationship of data in a specific time series. They applied their model on four equipment entities. In [15], the authors argue that new data sources generate daily a huge quantity of unstructured data. They conclude that to deal with such complex data, the use of big data analytic tools becomes an obligation. For that purpose, they propose a novel approach to detect and recognize containers code through a pure centralized approach.

Multi/trans-sectors: A flagship example is the watchdog principle developed in [16], see Fig. 4. As explained by the authors, this approach is generic and may

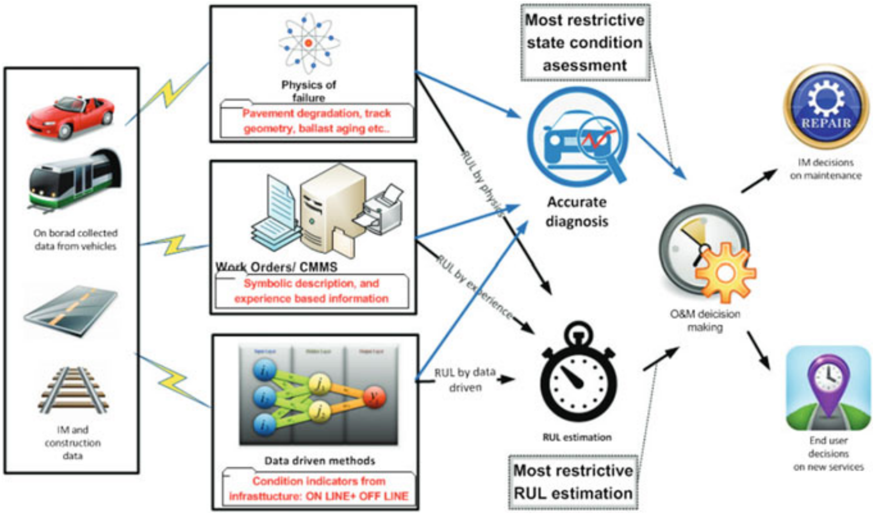


Fig. 3 Centralized architecture based on different information sources to estimate RUL [11]

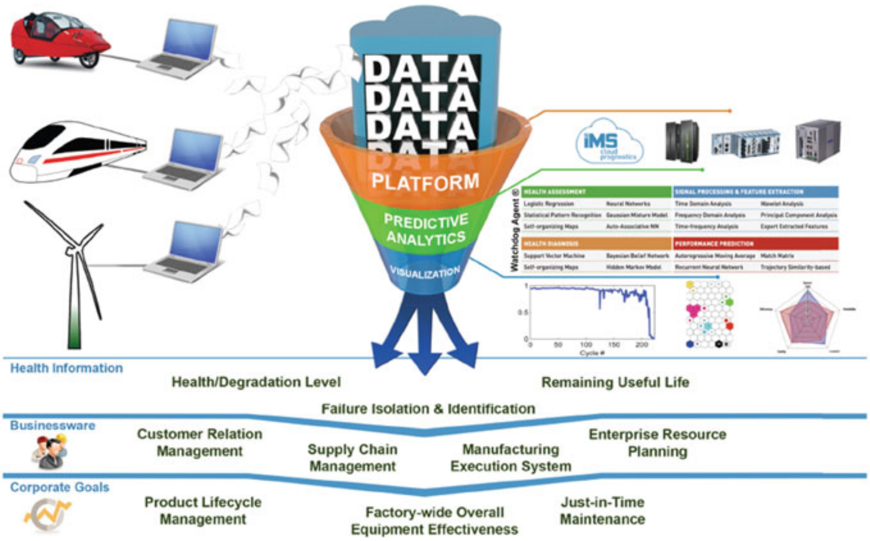


Fig. 4 The Watchdog principle [16]: a pure centralized approach

concern different systems and sectors, including the transportation one. The architecture is pure centralized. Advanced data analytics evaluates health information about the monitored systems. An interesting point in this approach is the necessary standardization of the data exchanges and KPI interfaces, which is, from our point of view, a necessary condition when seeking for an applicability in different industrial sectors, see [17]. Another similar approach focusing on prognostic can be found in [18]. In [19], the authors proposed a centralized architecture for the US Navy fleet management, with an interesting focus on data management. The proposed architecture integrates analytical solutions with event-driven monitoring of assets. The authors adopt a Model Driven Architecture (MDA) transforming sensor data to standardized data. The work describes a standard architecture, called OSA-CBM (Open System Architecture for Condition-Based Maintenance).

2.3 Synthesis from a Train Transportation Industrialist Point of View

From a train transportation industrialist point of view, the advantages of such pure centralized data management architectures are the following:

First, centralization of raw data enables the use of powerful data analytics/data mining methods to improve availability and equipment maintainability. This kind of architecture gains from the use of smart sensors and emerging technologies such as IoT. Big data models and methods are of course highly compliant with this kind of architecture with centralized intelligence; algorithms are centralized and address directly the set of raw data.

Second, one can deploy efficient massive data analytics, facilitating the identification of cause/effect relationships from sensor values to health status evaluation. A strong point is that these causalities relate to statistical confidence levels, which render them reliable. It is possible to modify or fine-tune expert knowledge according to these analytical studies.

Another strong advantage is that the data analytics models can identify some unexpected causalities since these models use a minimal “a priori” knowledge from experts. These new causalities can then foster the experts to reconsider and update their own knowledge, breaking old habits or rules for example.

Meanwhile, from the point of view a train transportation industrialist, pure centralized data management architectures suffer from several limitations.

First, associated data treatment methods, big-data analytics and others, require a stable set of data and need time to operate and be studied by the human operators. This is a long term, time-independent process that limits its exclusive use in industrial sectors for which a constant streaming of data flow can be faced as well as a need for short reaction time (for example, in case of a repetitive recent and not understandable set of similar failures occurring on different trains during their use).

Second, the centralized treatment of data implies the use of uniformed data while they are coming from complex moving systems, thus from various sensors, from various physical systems with various ranges, natures, base times and time stamps. Thus, there exists often a lot of necessary “craftwork” to prepare data: data normalization, cleaning, noise reduction or filtering processes to name a few. This craftwork may introduce biases in the data treatment algorithms whose impacts are currently hardly estimable. This "craftwork" approach also applies to the design of algorithms and models themselves since they still lack genericity and maturity: they remain highly specific to an equipment to be monitored (a motor, a track, a container…), highly dependent of the approach chosen by researchers for data analysis, thus hardly applicable to another equipment, limiting its acceptability by industrial engineers. Most of references reviewed address specifically one kind of equipment with no easy generalization to other ones.

Third, data is hardly associated with the exact past context in which they it was generated, which limits complementary latter in-depth exploration of causalities by human or legal experts. More, since transportation systems are mobile, there are often communication issues with the central off board system. In addition, since the moving systems have limited data storage capacities, industrialists often face complete disappearance of large sets of data because of embedded memorization capacity overload.

Four, concerning the architecture itself, pure centralized data management architectures are static and monolithic. Each minor modification is then costly. For example, it may be hard to insert a new kind of data or health status indicator in the global treatment. This is often a need when one understands that a fleet of transportation systems naturally evolves with time (insertion of a new set of trains/planes/cars/buses in a fleet, retrofit of adjustment of parts, overhaul of equipment, etc.).

Five, the complexity of the centralized data treatment algorithms implies that the computer developments are only manageable by expert people in data analytics while industrialists in transportation are still up to now, not expert in that field. Their knowledge and skills are more related to the transportation systems itself. This is a conflicting situation since these data are strategic for the industrialist and they obviously do not want to lose their control on them.

As a conclusion and to summarize, industrialists in the train transportation sector hardly face the 5V issues of big data [20] induced by the architecture itself, aiming to centralize as much as possible every possible raw data. From our point of view, initially deployed in a more stable context where time is not really essential (marketing, e-commerce, social network, biology, gaming, tourism, …), the use of pure centralized architecture requiring in depth data analytics in a big data environment meets its limits in industrial sectors facing a constantly changing environment, with hard time-based reaction constraints. Data analytics need time and stability to be complete, which is impossible in the train transportation sector when focusing on the availability and the maintainability of a fleet of trains in use, conveying people or hazardous material. A risk, faced actually by some of the industrialists when adopting such kind of architectures concerns the development of inefficient and costly mechanisms such as the "*store and forget*" one: terabytes of data are stored on hard-drives or in the cloud, waiting for a hypothetical late treatment.

3 The Mixed Centralized/Flat Hierarchical Approach

3.1 Definition

This approach consists in reducing the complexity of data management by isolating in the architecture independent physical systems or equipment and by associating to each of them a local *data management entity*, as an intermediary and elementary computing device of the architecture in charge of one or several local functions (e.g., monitoring). In this approach, these data management entities are directly dependent on a centralized data system and may gain from the recent emergence of the concept of cyber-physical systems. Such an approach can be optionally related to the previous one, enabling the use of data analytics, as depicted in Fig. 5. This architecture is denoted partially 'flat' in the sense that an equipment is associated to a data management entity for which there is no other hierarchical/multi-level structuration in this association.

3.2 Contributions

Literature dealing with this kind of architecture is more scarce but growing. Industrialists and researchers begin to recognize results from relevant contributions and potential benefits of this kind of architecture.

Manufacturing and logistics sectors: A typical example is provided by Lee and Bagheri [21], see Fig. 6. In this example, one can note that big data analytics

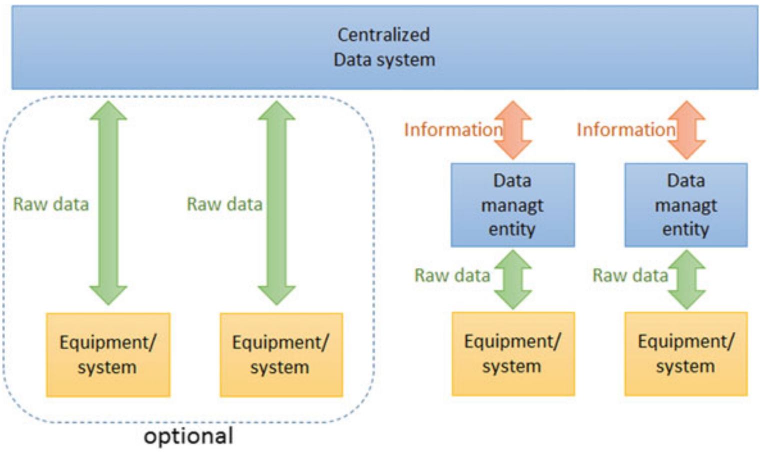


Fig. 5 The mixed centralized/flat hierarchical approach for developing data management

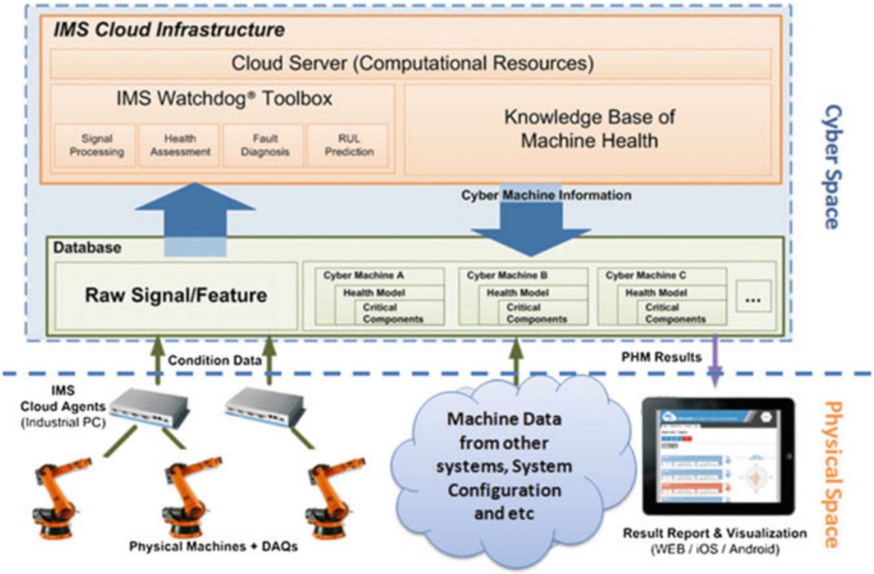


Fig. 6 A mixed centralized/flat hierarchical approach [21]

complete the use of the cyber-physical system approach, with centralization in a cloud infrastructure. The data management entities are digital avatars of their monitored equipment. This contribution translates a possible evolution towards future maintenance systems.

Aviation/aircraft sector: Another relevant contribution is suggested by Andreacchio et al. [22] with application to the maintenance of assets in a plane, see Fig. 7. In their work, which does not use the optional part of the approach, the authors suggested a cyber-physical approach coupled with RFID technologies to optimize jointly the preventive and the corrective maintenance processes. Data management entities are digital avatars of parts. They can emit locally information about their status according to plane operators' specifications facilitating their replacement in an efficient and reactive way.

A last illustrative example is the CPS4MRO architecture presented in [23], see Fig. 8. This architecture, dedicated to the maintenance, repair and overhaul (MRO) of aircraft equipment, is flat hierarchical. There is no centralization process of raw data. Informations associated to a data management entity are handled through a middle ware service bus. In this architecture, each part, tool, assets and units is considered either as an "intelligent" CPS or as a "simple" CPS. If the data management entity is an intelligent CPS, then it is associated to a digital avatar in a centralized remote agent platform.

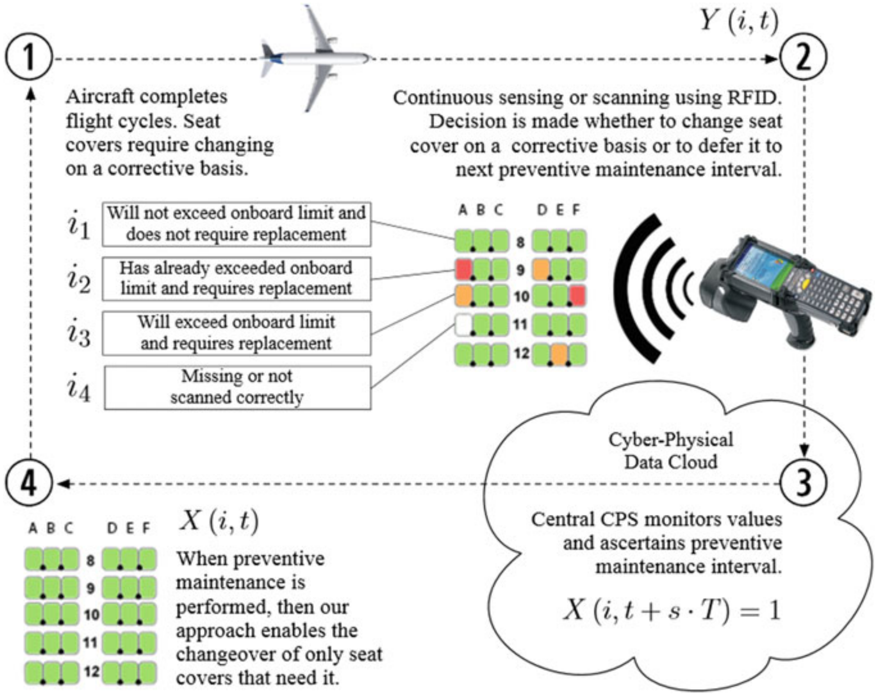


Fig. 7 Balancing predictive and corrective maintenance of aircraft assets using a flat hierarchical approach [22]

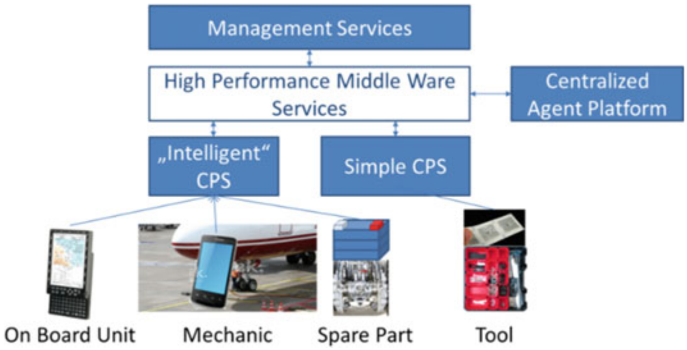


Fig. 8 The CPS4MRO architecture [23]: a flat hierarchical approach

3.3 Synthesis from a Train Transportation Industrialist Point of View

The common principle of the contributions relevant to this approach is to go a step further with the use of the concept of cyber-physical systems. This is done creating a kind of intermediary specific data management entity, which can be seen as a digital avatar/twin of physical equipment, able to integrate human expert knowledge about the systems whose availability and maintainability must be managed. This design principle can be seen as a first attempt towards the mirroring of some physical parts into a digital world [24], constructed using expert knowledge. Contributions relevant to this approach are not contradictory with the use of big data analytics, but try to integrate more knowledge from experts and tries a first "pre-structuration" of the data management process following the nature and the internal organization of the transportation system.

From a train transportation industrialist point of view, the advantages of mixed centralized/flat hierarchical management architectures are the following.

First, this kind of architecture suggests a smarter use of the knowledge about the system behaviour and a decoupling of data (strong reduction of combinatorial complexity) using the principle of an avatar/twin, which simplifies the data treatment time by decoupling a priori un-coupled data.

Second, and compared to the previous approach, this approach can be more easily handled by train transportation operators and constructors since the avatar/twin can embed rules from train transportation experts dealing with the associated physical component.

Third, this architecture is compatible with data centralization and novel computer network paradigms, such as edge computing, thus complementary use of previously introduced data analytics is feasible, under the condition that raw data gathering processes are designed along with the set of avatars/twins.

Four, an interesting aspect is that different kinds of systems can interoperate through the use of avatars, outside the transportation system itself (e.g., tool, spare part…), opening possibilities to inform maintenance centers for example.

Meanwhile, mixed centralized/flat hierarchical architectures suffer from several limitations when applying them in train transportation:

First, since the architecture is flat, it works well for dedicated systems or equipment (robots, machine, part, tool…) considered as an "atomic", entity-as-a-whole system, but not for complex integrated multi-level systems such as trains, cars or planes.

Second, one can still face a lack of genericity in the design of the architecture in the literature and relevant algorithms remain highly application-dependent and equipment-dependent.

Third, there are some difficulties to design and integrate in this kind of architecture mechanisms ensuring the robustness of results in real time (e.g., difficulties to handle alarm bursts, false alarms, search for causalities, etc.).

Last, the basic use of avatars/twins is encouraging and innovative, but these digital entities still remain passive, with few decisional abilities coupled with still a high level of centralization (e.g., in the cloud), limiting the potential benefits of local reactive intelligence and emergence of new knowledge constructed through direct interaction and cooperation among the avatars/twins.

4 The Semi-heterarchical Approach

4.1 Definition

In this approach, a set of hierarchized data management entities are associated to a subsystem of the transportation system according to a desired level of granularity, from sensors, components, equipment to the transportation system itself or even its fleet, see Fig. 9. These entities are associated, with respect to their level of granularity, to a physical counterpart for which they fulfil a set of functions (e.g., raw data memorizing, monitoring, diagnosis, RUL estimation, etc.) contributing to the estimation and the improvement of the availability and the maintainability of the whole system. The hierarchical decomposition of a system into sub-systems applies: a system at an upper level contains the system of the lower levels, as depicted in Fig. 9.

4.2 Contributions

Despite its scarceness in the transportation sector, one can identify some illustrative contributions in other fields, mainly in manufacturing and logistics.

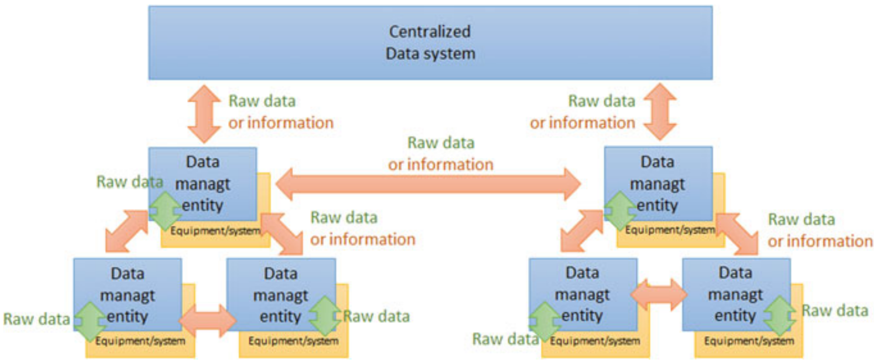


Fig. 9 The semi-heterarchical approach for designing data management

Manufacturing and logistics sectors: In these sectors, this architecture has been widely studied. Favoured modelling approaches are the multi agent and the holonic paradigms [25]. The principle of recursivity (a holon is a whole, composed of smaller holons and a part as well, composing a greater holon) and the mirroring of the reality (a holon is composed of a physical and a digital part) [26], which is compatible with the systemic decomposition principle are interesting aspects of the holonic paradigm. Examples in manufacturing are provided in [27] and in logistics in [28, 29] and [30]. In [29], the authors propose an holonic architecture to road management system with a focus on specific issues such as congestion and accidents. In [30], a holonic architecture is proposed to compute the shortest path in complex road transportation networks.

Rail sector: In [31], a holarchy dedicated to the monitoring of train door was proposed for the first time, see Fig. 10. The proposed application gave interesting results from real experiments. Meanwhile, in their work, [31] tested only basic dedicated rules embedded recursively into holons (mainly, fusion of similar events and inhibition of false alarms) to help the monitoring of doors, but with no seek for real genericity. More, the fleet level was not addressed.

In [32] the ISO 13374 standard for monitoring and diagnosis served as a basis to specify the generic data treatment process of the holon modelled by Le Mortellec et al. [31] into a train and a new kind of holon, in the wayside maintenance centre. This process ranges from data acquisition to prognostic assessment. This work can be considered as a first extension of the work of [31] toward the specification of a fleet management system using holonic principles.

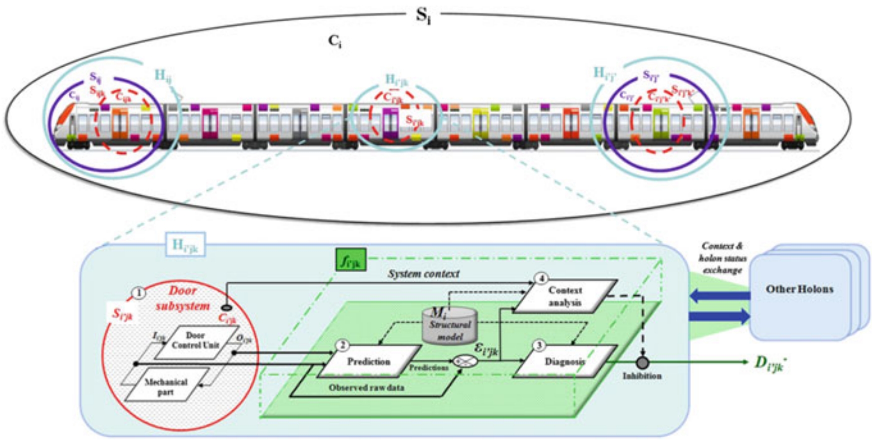


Fig. 10 A holonic model of a train with application to door monitoring [31]

4.3 Synthesis from a Train Transportation Industrialist Point of View

From a train transportation industrialist point of view, the main strong point of this third approach is that it solves the major drawbacks induced by the previously presented second approach.

First, this kind of architecture suits well the context of complex engineered systems: it can be viewed as a recursive integration of smaller interacting systems, as it is the case in train transportation. This aspect solves the corresponding issue associated to the second architecting approach.

Second, it widens the possibilities offered by the second approach through a more elaborated architecture, based on more active, intelligent and cooperative data management entities, facilitating the handling of robust mechanisms contributing to the real improvement of the global availability and maintainability of the train transportation systems and their fleet.

In addition, it is worth mentioning that this kind of architecture does not feature centralization mechanisms enabling data analytics and the use of big data. But such integration is meanwhile possible: data analytics (big data) approach is still feasible if the designer integrates in his architecture not only the information and cooperation mechanisms among holons/agents, but also functions for gathering and transmitting raw data from sensors from the bottom of the architecture to the top centralized data management system.

In fact, the main drawback of such an approach relies on the scarceness of the existing literature in the transportation domain compared to other domains such as manufacturing and logistics. Consequently, one can face a clear lack of generic models and methods dedicated to the definition of the architecture of the third kind with applications to a fleet of trains.

5 Challenges for the Near Future

From our review, some challenges can be identified for the near future since some major gap with industrial and societal needs are still not filled. Among these challenges, let us mention, whatever the architecture:

- The optimisation of the embedded data management devices in terms of reliability, memory peak needs, energy consumption and communication bandwidth (which can be denoted as “embedability”)
- The development of dependability analysis applied to the data management architecture itself (according to the adage: “Quis custodiet ipsos custodes?”) to avoid the paradox where the architecture aiming to improve availability actually reduces it because of generation of false alarms, failure of the added data management devices, etc.

- The improvement of the designed architecture to enable dynamic optimization of the maintenance strategy of the fleet
- The search for a generic applicability of the architecture, independent from the initial application field, to any kind of fleet, accompanied with a deployment methodology including risks analysis, deployment cost analysis and evaluation of the return of investment (ROI)
- The design of ergonomic human-centred interfaces with the architecture, including support systems using the new information generated for the maintainers, engineers and fleet operators
- Given the long lifespan of considered systems, the modelling and capitalization of the knowledge between successive projects is required to improve the different versions of the data management architectures, considering the rapid technological evolution and the obsolescence of past deployments
- The search for integration with other services and department, for example with Integrated Logistics Support (ILS) and design system engineering since these novel data management approaches will imply the creation for new business, jobs and skills
- Aligned with the previous challenge, the search for the interoperability with other enterprise information systems (PLM, etc.).

6 Conclusion and Future Works

This chapter contains a literature review regarding the different ways a data management architecture can be designed in transportation, manufacturing and logistics sectors to improve the availability and maintainability of an equipment, a system or a fleet. For that purpose, three approaches have been identified, characterized and evaluated from the point of view of an industrialist working in the train transportation sector. Each of these architectures presents advantages and drawbacks that have been discussed. Meanwhile, from our point of view, these advantages and drawbacks must obviously be studied according to the specific context expressed by the industrialist aiming to deploy such a kind of architecture. This specific context concerns the international market, the ambitions of the competitors, the technological offer, the complexity of the systems composing the fleet, the national incentive and legal policies, the norms and the expectations of operators and end users.

The next step of our work is to choose one specific architecture and to develop an original data management system based on this architecture with the goal to solve some of the introduced challenges with application to a fleet of trains. This work has thus to be based on a context (that remained to be pointed out) and on the analysis led in this chapter. This original data management system is then detailed in [33].

Acknowledgements This work is done in the context of the joint research Lab, "Surferlab" (http://www.surferlab.fr/en/home), founded by Bombardier Transport, Prosyst and the University of Valenciennes and Hainaut-Cambrésis. SurferLab is scientifically supported by the CNRS and is partially funded by ERDF (European Regional Development Fund). The authors would like to thank the CNRS, the European Union and the Hauts-de-France region for their support.

References

1. Voisin, A., Medina-Oliva, G., Monnin, M., Leger, J.-B., Iung, B.: Fault diagnosis system based on ontology for fleet case reused. In: Ebrahimipour, V., Yacout, S. (eds.) Ontology Modeling in Physical Asset Integrity Management, pp. 133–169. Springer International Publishing (2015)
2. Simon, A.: Modern Enterprise Business Intelligence and Data Management. Morgan Kaufmann (2014)
3. Trentesaux, D.: Distributed control of production systems. Eng. Appl. Artif. Intell. **22**, 971–978 (2009)
4. Chen, J., Lyu, Z., Liu, Y., Huang, J., Zhang, G., Wang, J., Chen, X.: A big data analysis and application platform for civil aircraft health management. In: 2016 IEEE Second International Conference on Multimedia Big Data (BigMM), pp. 404–409 (2016)
5. Yang, L., Wang, J., Zhang, G.: Aviation PHM system research framework based on PHM big data center. In: 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), pp. 1–5 (2016)
6. Larsen, T.: Cross-platform aviation analytics using big-data methods. In: 2013 Integrated Communications, Navigation and Surveillance Conference (ICNS), pp. 1–9 (2013)
7. Núñez, A., Hendriks, J., Li, Z., Schutter, B.D., Dollevoet, R: Facilitating maintenance decisions on the Dutch railways using big data: The ABA case study. In: 2014 IEEE International Conference on Big Data (Big Data), pp. 48–53 (2014)
8. Fumeo, E., Oneto, L., Anguita, D.: Condition based maintenance in railway transportation systems based on big data streaming analysis. Procedia Comput. Sci. **53**, 437–446 (2015)
9. Capasso, M., Savio, S., Firpo, P., Parisi, G.: Railway transportation efficiency improvement: loco health assessment by time domain data analysis to support condition based maintenance implementation. Transp. Res. Procedia **6**, 424–433 (2015)
10. Sammouri, W.: Data mining of temporal sequences for the prediction of infrequent failure events : application on floating train data for predictive maintenance, https://tel.archives-ouvertes.fr/tel-01133709/document (2014)
11. Thaduri, A., Galar, D., Kumar, U.: Railway assets: a potential domain for big data analytics. Procedia Comput. Sci. **53**, 457–467 (2015)
12. Shi, Q., Abdel-Aty, M.: Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways. Transp. Res. Part C Emerg. Technol. **58** (Part B), pp. 380–394 (2015)
13. Wang, H., Osen, O.L., Li, G., Li, W., Dai, H.-N., Zeng, W.: Big data and industrial Internet of Things for the maritime industry in Northwestern Norway. In: TENCON 2015—2015 IEEE Region 10 Conference, pp. 1–5 (2015)
14. Yuanyuan, L., Jiang, S.: Research on equipment predictive maintenance strategy based on big data technology. In: 2015 International Conference on Intelligent Transportation, Big Data and Smart City, pp. 641–644 (2015)
15. Ayed, A.B., Halima, M.B., Alimi, A.M.: Big data analytics for logistics and transportation. In: 2015 4th International Conference on Advanced Logistics and Transport (ICALT). pp. 311–316 (2015)
16. Lee, J., Ardakani, H.D., Yang, S., Bagheri, B.: Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. Procedia CIRP **38**, 3–7 (2015)

17. Lee, T., Tso, M.: A universal sensor data platform modelled for realtime asset condition surveillance and big data analytics for railway systems: developing a Smart Railway mastermind for the betterment of reliability, availability, maintainability and safety of railway systems and passenger service. In: 2016 IEEE Sensors, pp. 1–3 (2016)
18. Meraghni, S., Terrissa, L.S., Ayada, S., Zerhouni, N., Varnier, C.: Post-prognostics decision based on cyber-physical systems. In: 12ème Congrès International de Génie Industriel, p. #25, Compiègne, France (2017)
19. Tambe, S., Bayoumi, A-.M.E., Cao, A., McCaslin, R., Edwards, T.: An Extensible CBM Architecture for Naval Fleet Maintenance Using Open Standards, Presented at the Intelligent Ship Symposium, Boston, USA (2015)
20. Yin, S., Kaynak, O.: Big data for modern industry: challenges and trends [Point of View]. Proc. IEEE **103**, 143–146 (2015)
21. Lee, J., Bagheri, B.: Cyber-physical systems in future maintenance. In: Amadi-Echendu, J., Hoohlo, C., Mathew, J. (eds.) 9th WCEAM Research Papers, pp. 299–305. Springer International Publishing (2015)
22. Andreacchio, M., Bekrar, A., Benmansour, R., Trentesaux, D.: Balancing preventive and corrective maintenance of aircraft assets: a cyber-physical systems approach. In: 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), pp. 500–503 (2016)
23. Trentesaux, D., Knothe, T., Branger, G., Fischer, K.: Planning and control of maintenance, repair and overhaul operations of a fleet of complex transportation systems: a cyber-physical system approach. In: Service Orientation in Holonic and Multi-agent Manufacturing, Studies in Computational Intelligence, vol. 694, pp. 175–186. Springer International Publishing (2015)
24. Rutten, L., Valckenaers, P.: Self-organizing prediction in smart grids through delegate multi-agent systems. In: Pérez, J.B., Rodríguez, J.M.C., Fähndrich, J., Mathieu, P., Campbell, A., Suarez-Figueroa, M.C., Ortega, A., Adam, E., Navarro, E., Hermoso, R., Moreno, M.N. (eds.) Trends in Practical Applications of Agents and Multiagent Systems, pp. 95–102. Springer International Publishing (2013)
25. Barbosa, J., Leitão, P., Adam, E., Trentesaux, D.: Dynamic self-organization in holonic multi-agent manufacturing systems: the ADACOR evolution. Comput. Ind. **66**, 99–111 (2015)
26. Koestler, A.: The ghost in the machine. Hutchinson, London (1967)
27. Zambrano Rey, G., Pach, C., Aissani, N., Bekrar, A., Berger, T., Trentesaux, D.: The control of myopic behavior in semi-heterarchical production systems: a holonic framework. Eng. Appl. Artif. Intell. **26**, 800–817 (2013)
28. Sallez, Y., Berger, T., Raileanu, S., Chaabane, S., Trentesaux, D.: Semi-heterarchical control of FMS: from theory to application. Eng. Appl. Artif. Intell. **23**, 1314–1326 (2010)
29. Versteegh, F., Salido, M.A., Giret, A.: A holonic architecture for the global road transportation system. J. Intell. Manuf. **21**, 133–144 (2010)
30. Beckers, C.H.J.: A holonic architecture for the global automated transportation system (GATS), http://essay.utwente.nl/59154/
31. Le Mortellec, A., Clarhaut, J., Sallez, Y., Berger, T., Trentesaux, D.: Embedded holonic fault diagnosis of complex transportation systems. Eng. Appl. Artif. Intell. **26**, 227–240 (2013)
32. Fadil, A., Clarhaut, J., Branger, G., Trentesaux, D.: Smart condition based maintenance (S-CBM) for a fleet of mobile entities. Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in Computational Intelligence, vol. 694, pp. 115–123. Springer, Cham (2017)
33. Trentesaux, D., Branger, G.: Foundation of the Surfer Data Management Architecture and its application to train transportation. In: Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in computational Intelligence, vol. 762. Springer (2018)

Foundation of the Surfer Data Management Architecture and Its Application to Train Transportation

Damien Trentesaux and Guillaume Branger

Abstract Data management architectures are key elements to support the improvement of the availability and the maintainability of fleets of transportation systems such as trains, cars, planes and boats during their use. In this context, this chapter proposes the foundations of a specific data management architecture, named "Surfer". The design of this architecture follows a set of specifications, named "the Surfer way" that translates an original way to consider the issue of big data, aiming to transform raw data into high level knowledge usable by engineers and managers. An application of the Surfer architecture to train transportation is presented. First results are encouraging, the train constructor expects a gain up to 2% of the availability of a fleet of trains.

Keywords Data architecture · Data management · Availability
Maintainability · Train transportation

1 Introduction

This chapter deals with the way constructors of a fleet of complex systems, and especially in the field of transportation (road, rail, aircraft and naval sectors) can improve the performances of their fleet during their use, the fleet being operated by a public operator or a private company. In that context, data management architectures are key elements to support the improvement of the availability and the maintainability of such fleets. The paper proposes the foundations of a specific data management architecture, named "Surfer".

D. Trentesaux (✉)
LAMIH UMR CNRS 8201, SurferLab, University Valenciennes,
Le Mont Houy, 59313 Valenciennes Cedex Valenciennes, France
e-mail: damien.trentesaux@univ-valenciennes.fr

G. Branger
Bombardier Transport, 1 Place Des Ateliers, 59154 Crespin, France
e-mail: guillaume.branger@rail.bombardier.com

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_9

The idea to propose a new architecture comes from an observation of the current context that can be characterized by the identification of two kinds of forces and several complexity factors driving the evolution of constructors' products and conditioning their international competitiveness. The first kind of forces, denoted here the "**pull forces**", translates the evolution from the demand side. From this point of view, at national (politics, national operators, etc.) and local (logistician, etc.) levels, one can face new expectations (economic, societal and environmental) federating numerous key-performances indicators (KPI) expressed in terms of manufacturability, lifecycle cost, environmental footprint, recyclability, availability and security to name a few. For example, the guarantee of a high level of dependability of truck or train urban transportation systems is a critical stake because of the fact that urban infrastructures are saturated, impossible to extend in dense urban areas. The huge financial penalties involved when this level of availability is not met during the exploitation of a fleet of transportation systems translates this stake. More, operators require not only available fleets, but also advanced exploitation services along with the fleet, typically, monitoring and maintenance services, leading constructors to sell more and more transportation services with high expectation from operators and users instead of selling physical transportation systems alone.

Different factors make it hard for the constructors to meet these new expectations expressed by these "pull forces", denoted here "**complexity factors**". We provide hereinafter a non-exhaustive list of some of the main ones:

First, transportation systems are complex engineered systems in the sense that they integrate equipment with multi-physics technological solutions, coming from different engineering fields (mechanical engineering, computer and control sciences, electronics, thermal engineering, etc.). Equipment behaviours and data generated from sensors are thus diversified, which makes it difficult to handle them in an integrated and consistent way [1]. Experts of various scientific and technological fields must characterize in a correct manner each equipment.

Second, transportation systems are mobile elements evolving in uncontrolled open environments, which complicate the expertise in case of failure, the reactive optimization of maintenance processes and the management of spare parts, operator skills and maintenance resources.

Third, transportation systems, and especially trains, planes and ships are long lasting systems compared to the rapid evolution of the introduced "pull forces" coupled with the one of the technological offers (see the push forces hereinafter). This makes it hard for the constructors to put on the market products and transportation services that are able to adapt or to be adapted to these evolutions. More, and specifically for the train transportation sector, the whole set of fleets is aging, with less and less new contracts, implying the rise of the importance of Maintenance, Repair and Overhaul (MRO) operations stakes and costs.

Fourth, a fleet of transportation systems is often composed of systems characterized by histories highly different one from the others, with different historical MRO operations coupled with different use rates and usages, depending, in addition, of

the type of climate and of the geography where the mobile system evolves. It is also worth mentioning that the composition of a fleet itself generally evolves significantly, given the life span of the considered systems, from early product introductions in the fleet to the late ones, potentially several years later [2].
Fifth, the warrantee situations and conditions evolve as well with time (constructors' warrantees are provided for a limited amount of time) for each element of the fleet, strongly influencing policies, behaviours of stakeholders and maintenance decisions [2].
Sixth, the human aspect is of great importance for constructors and operators. The existence of various human based organizations that interact is a critical factor: operators' organization as well as constructors' one and users' community are fundamentally human-based and often scattered at a regional or a national level, based on dedicated or highly specialized skills, with strong historical habits and sometimes, a clear reluctance to evolve. More, human decisions about a fleet must be taken in an unstructured decisional environment, with limited reliable information and conflicting objectives, which makes the balanced and effective management of a fleet of transportation systems hard to obtain.
And seventh, since a fleet is composed of relatively autonomous and decoupled but linked mobile entities on the same infrastructure, a fleet can be considered as a "system of systems", which is known to be reactively and proactively managed with difficulty [3].

The second kind of forces is denoted "**push forces**". These forces translate the existence of disruptive recent technological offers that can be considered as opportunities by constructors to meet the expectations imposed by the "pull forces" while handling at the same time the "complexity factors". This technological offer finds its origin in recent advances in various scientific fields such as mechanical engineering (e.g., lightweight car), thermal engineering (e.g., low energy train) or in the information and communication technologies (ICT) field encompassing electronics, computer science and control science. Specifically, and relevant to this ICT field, three main kinds of contributions are currently highly studied by researchers:

The first one relates to the recent development of *innovative models and methods in artificial intelligence* (AI) techniques. Relevant contributions includes for example recent advances in supervised or non-supervised learning capabilities such as neural networks, deep learning and deep belief [4], image recognition and pattern clustering [5] to name a few.
The second type is related to the recent development of *innovative models and methods for MRO*, potentially using previously introduced advances in AI, and including works on Prognostic and Health Management (PHM), Integrated Systems Health Management (ISHM) and remaining useful life (RUL) estimation [6]. These developments lead typically to contributions to condition based maintenance (CBM), smart maintenance, recyclability and deconstruction, etc. with a growing number of applications in the transportation sector, e.g., [5, 7, 8]. These contributions typically include recent advances in fault diagnosis methods, being

data-based or model-based, such as big data analytics [9], Support Vector Machines [10] or Bayesian networks [11].
The third one is related to contributions on innovative kinds of *fleet data management architectures*, potentially embedding the two previous kinds of advances, and based on the use of Internet of Things (IoT) [12], smart sensors [13], embedded intelligence [14], and cyber-physical systems (CPS) [15]. These contributions include for example recent advances in multi-agent and holonic modelling approaches [16].

To summarize, these “push forces” help meeting the expectations federated through the “pull forces” and simultaneously provide solutions to the introduced “complexity factors”, as summed up in Fig. 1. In this figure, a feedback loop is suggested, closing the loop from the “pull forces” when expectations are not met by products and services, to influence the evolution of “push forces”. This loop, not studied in this paper, must be kept in mind when developing new solutions to be put on the market or adapting existing ones.

In this context, our work focuses more precisely on a subset of the introduced forces:

- “Pull forces”: we address here mainly two (but closely related) KPI which are Availability and Maintainability of a fleet of transportation systems and the way they can be managed and improved in real time. Availability means “the readiness for correct service” while Maintainability means “the ability to undergo repairs and modifications”, both of them representing one aspect of the broader concept of dependability [3].

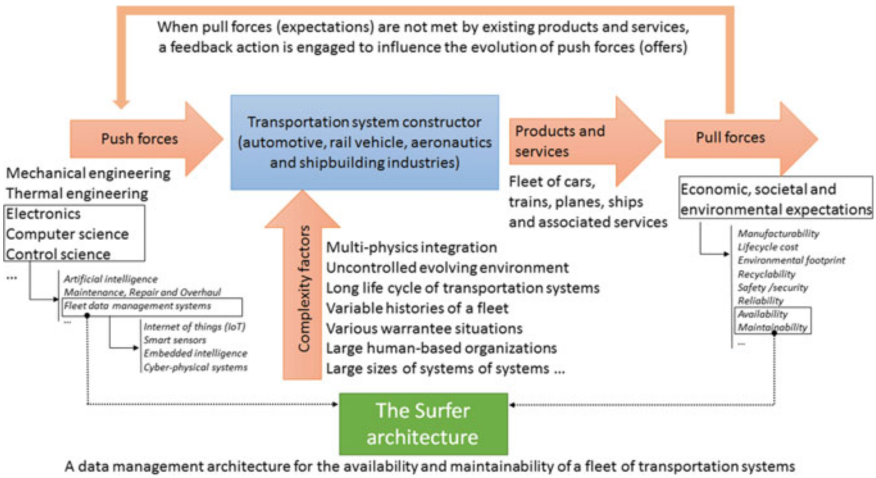


Fig. 1 Push forces, pull forces and complexity factors for transportation system constructors

- "Push forces": we address here only the third kind of "push forces" related to the development of "fleet data management systems". It is thus important to note that contributions relevant to this third kind can be dependent or not of one or several modelling approaches relevant to the two other ICT kinds of push forces, namely innovative models and methods in AI and for MRO.

In this context, our work is devoted to the definition of an original fleet data management system to help transportation system constructors and operators to manage and improve the availability and the maintainability of their fleet during their use. In this chapter, we present an original data management architecture, called the "Surfer Architecture" which is positioned in Fig. 1. Relevant design choices have been carefully elaborated from a previous review of the literature [17] coupled with a fine-tuned evaluation of the context in the domain of train transportation. The aim of this paper is to present the foundations of this original fleet data management architecture and one of its applications to train transportation.

The outline of the paper is the following: in Sect. 2, we present the foundations of the Surfer architecture while in Sect. 3 an application to train transportation, named TrainSurfer is introduced. This paper concludes with a discussion about the implementation of the TrainSurfer architecture.

2 Foundations of the Surfer Architecture

In this section, we specify the Surfer architecture in a generic way, targeting a set of possible applications on different kinds of transportation systems. Following the conclusions of a review of the different kinds of architecture [17], the suggested Surfer Architecture has been designed as a generic holonic semi-heterarchical [18] data management architecture. In [17]:

A *data management system* is an information system aiming to gather, memorize, manipulate and communicate data, information and digitalized knowledge coming from physical equipment and systems to users, managers or other information systems of a company.
A *data management architecture* describes the functions and algorithms (software) to be supported as well as the localization of the different computing devices (hardware) in charge of the flow of data, information and digitalized knowledge of a data management system.

The Surfer Architecture is a specific data management architecture aiming to improve the availability and maintainability of a fleet of transportation systems with application to different transportation sectors. The Surfer architecture can also be seen as an extension and a generalization of previous works presented in [19] and [20].

2.1 Fields of Valid Application

The Surfer Architecture requires some assumptions about the type of target systems on which it is applied to ensure its consistent use:

The Surfer Architecture is to be applied on different mobile systems, each of them implementing a similar global function regarding transportation (ex: train, plane, etc.) and interacting with an infrastructure. Thus, these target systems are sufficiently homogeneous to enable full cross comparison to help the relative discrimination of events and data, as illustrated in [21].
From an internal point of view, each of these target systems can be viewed as a hierarchized integration and coupling of sub-systems, whatever their intrinsic nature (electrical, mechanical…). For each of these sub-systems, a model of their normal behaviour is available at the desired degree of precision to enable current and future discrimination of normal and abnormal situation.
From an external point of view, each of these target systems merges into a fleet. They are sufficiently independent from each other to enable some decoupling in their management.

Obviously, ground, maritime and aerial transportation fields are relevant application domains, but from these assumptions one can note that other application fields are feasible (e.g., fleet of mobile robots, AGV, PRT, etc.).

2.2 The "Surfer Way"

The fundamental design principles regarding the Surfer Architecture are the following:

The "everybody is holon" view applies [16], every equipment, up to the entire mobile system is seen as holon. There are artificial holons (part, train), called *Surfer holons*, cooperating with *human holons*. Each artificial holon consists of the mirroring of a physical part by a digital counterpart (as the CPS approach suggests), the digital part potentially modelled using multi-agent approach.
The principle of recursivity in holonic architecture applies, that is the ability to reproduce a similar pattern (here, a data management process), from top level system (e.g., a plane or a fleet of planes) to the lowest desired level, defining the desired granularity by the designer (e.g., the equipment or the component).
The design of the Surfer Architecture must be done in such way that each application of the Surfer Architecture is human-centered, that means that every profile of possible end user must be considered from the beginning of the application process, as suggested in [22].

The main specifications following these design principles are presented hereinafter. These specifications are organized according to five categories: (1) Holonic architecture, (2) Surfer Holons, (3) Integration, (4) Data Structure, treatment and communication, and (5) Deployment by experts. They translate a certain vision of the architecture to design. We federate them under the wording "**the Surfer way**".

2.2.1 Holonic Architecture

The specifications regarding the Holonic architecture are the following:

The designed architecture, composed of Surfer Holons interacting with Human Holons, complies with the three introduced holonic principles and is structured as a semi-heterarchical one, see [17].
The Surfer architecture is an open architecture in the sense that nothing is assumed about any design choice related to the two other pull forces in ICT, such as innovative models and methods in AI and for MRO. The Surfer architecture can integrate various contributions from these research fields. It requires only the integration of different low-level technological solutions for sensing, data generation/communication (network interface), data analysis methods or more elaborated state-of-the-art diagnosis, prognostic and health-status technics, methods or models. Consequently, it requires the ability to read all the needed data through ad hoc networks and sensor interfaces. Through this interface, data enter the "Surfer world" and are all handled in the same way, whatever the level considered (low level: component/part, high-level: fleet), see the set of specifications related to data structure, treatment and method.
The designed architecture must be non-intrusive, i.e., it should not influence the security or the reliability of the transportation system into which it is integrated, thus it does not require any SIL (safety integrity level) studies.

2.2.2 Surfer Holons

The specifications regarding the Surfer Holons are the following:

Each Surfer Holon mirrors a physical counterpart to be monitored (see [17]). The level in the architecture defines the level of integration of the couple (entity, system to monitor). It ranges from low-level sensor (a Surfer Holon monitors a sensor) to high-level equipment (a Surfer Holon monitors a HVAC) and even the transportation system itself (a Surfer Holon monitors a plane). Monitoring its associated physical system behaviour is the basic function of a Surfer Holon. More elaborated functions can be designed dealing for example with diagnosis, prognostic or evaluation of health status (PHM) or remaining useful life (RUL) of the associated system.

Surfer Holons composing the architecture are active, that is at least able to memorize and filter data, trigger events and generate alarms when monitoring their physical systems. More intelligent abilities can be developed, including decisional abilities and abilities to collaborate with peers as well as the human, according to the known principles of an "active product" [23].
Surfer Holons own dedicated human-machine interface, more or less elaborated, the human being an engineer, a maintainer or a fleet supervisor, depending on the corresponding level of the holon. The design of these interfaces is human-centred.
Hardware implementations of Surfer Holons should be favoured in an embedded way, close to the systems they monitor to access localized contextualized data (the closest, the better) but it is not compulsory if networking and remote data mirroring technologies enable off board implementations (e.g., edge computing).

2.2.3 Integration

The specifications regarding integration are the following:

- The surfer architecture must be interoperable with future MRO processes and systems, including e-maintenance, smart maintenance, dynamic maintenance, stealth maintenance, opportunistic maintenance, etc.
- The surfer architecture must be compatible with the "intelligence of everything" view, considering that, through the Surfer architecture, a transportation system is able to interact and collaborate with other active systems inside (e.g., other mobile systems, maintenance supervisors) or outside of the environment of the operator (e.g., the infrastructure, maintenance part suppliers).
- The surfer Architecture must enable the closing of the loop with design and manufacturing services of the constructor, back-forwarding knowledge from the exploitation and maintenance history of the transportation system to designers. This enables the enlarging of the scope of the study from the single "use phase" to encompass aspects relevant to the whole product lifecycle. The Surfer architecture should also facilitate the management of other kinds of feedback loop, and especially the one introduced in the Fig. 1 dealing with the adaptation of ICT "push forces" to better suit to unsatisfied "pull forces" expectations.

2.2.4 Data Structure, Treatment and Communication

The specifications regarding the data structure, treatment and communication are the following:

Each raw data entering the Surfer architecture must be translated into a standardized, interoperable, structured, and enriched format, called *Surfer Data*. The aim is to time, sample, localise and contextualize these raw data according to expert knowledge. A similar specification applies to events (called *Surfer Event*). Algorithms and rules are designed assuming such standardization. This is critical to help on-line and future expertise of faults and alarms, including false alarms as well as to ensure future possible interoperability with other information systems.

The surfer Architecture must address security and possible hacking issues. Meanwhile, since the architecture is not intrusive, this issue is not as fundamental as it could be if critical functions were concerned. Widening of this aspect within the challenging context of ethical behaviour of cyber-physical systems is one of the most critical aspect to address in the near future [24, 25].

Each Surfer Data or Surfer Event is processed by one or several Surfer Holons according to a set of *Surfer Rules*. These rules are either based on classical "if… then…" assertions or on mathematical equations aiming to elaborate indicators or on action rules aiming to trigger events.

The OSA-CBM (Open System Architecture for Condition-Based Maintenance) industrial standard is used as a first approach to define, discriminate and layer the role of each data, from sensors raw data to high-level decision. This aspect is dealt with in [19].

2.2.5 Deployment by Experts

The issue "how to deploy the Surfer Architecture on an existing fleet?" is not completely solved, since until now only one application in train transportation, presented hereinafter, is available. The elaboration of such a deployment method will be addressed later. The method must at least be constructed through the implementation of the previously introduced "Surfer items" (Surfer Data, Surfer Events, Surfer Rules, Surfer Holons and architecture) based on the knowledge coming from the following experts:

Constructor expert knowledge about the most accurate physical models of sub-systems enabling their monitoring and assessment of their health status and prognostic

Constructor expert knowledge about the propagation rule of events running on physical networks, wireless network and databases

Surfer Designer expert knowledge about the information process analysis in order to establish the list of holons and to suppress the unnecessary manual links and activities (e.g.,.csv files to sum up weekly report)

Surfer Designer expert knowledge to configure and deploy the architecture and the Surfer Holons (including their parameters) and to connect them to the network

Surfer Designer and Constructor expert knowledge about the evaluation of computational needs and capacities embedded in the mobile systems, including network bandwidth and cyber-security aspects.

3 Experimentation in Train Transportation: The TrainSurfer Architecture

We detail hereinafter an application of the Surfer Architecture to train transportation, named TrainSurfer Architecture. Returns from experience are discussed at the end of the section. The TrainSurfer architecture, as an application of the Surfer architecture, is semi-heterarchical, see Fig. 2.

Figure 3 presents a model of a TrainSurfer Holon. Following the idea proposed by Fadil et al. [20], there are two kinds of TrainSurfer holons: on-board (embedded in the transportation system) or off-board (on the infrastructure or in supervision or maintenance centres).

The digital part of the Holon is modelled as an agent and is called TrainSurfer agent. The TrainSurfer agent is responsible for the management of raw data from physical equipment and systems, at least from the monitored physical part but potentially from other physical systems. It sends and receives Surfer data to/from other TrainSurfer agents. It embeds a set of rules that apply to the Surfer data it receives and generates. These rules can be simple basic rules such as (in text form) "`compute the average current value for the 10 last opening/closing cycles for the monitored door`" or more complex ones such as "`if (value > v_threshold) and (average current > c_threshold) then send "Alarm1" to SurferAgent_2`". These rules come from expert knowledge. Figure 4 presents a screenshot of the designed TrainSurfer Agent rule editor.

Off –board TrainSurfer holons have the same structure as on-board ones but they are not associated to any physical system. They are in charge of the interaction between the on-board TrainSurfer holons and the "rest of the world": maintenance supervisors, maintenance planning/scheduling software, centralized databases (for big data analyses), etc. Up to now, the role of these Off-board TrainSurfer holons is

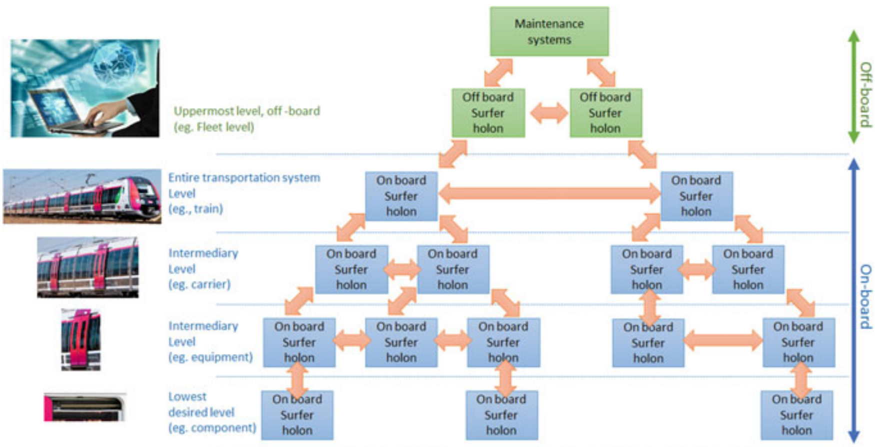


Fig. 2 The TrainSurfer architecture—a semi-heterarchical data management architecture

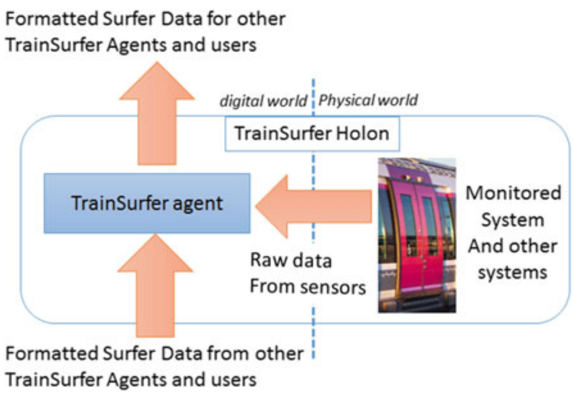


Fig. 3 Model of a TrainSurfer holon

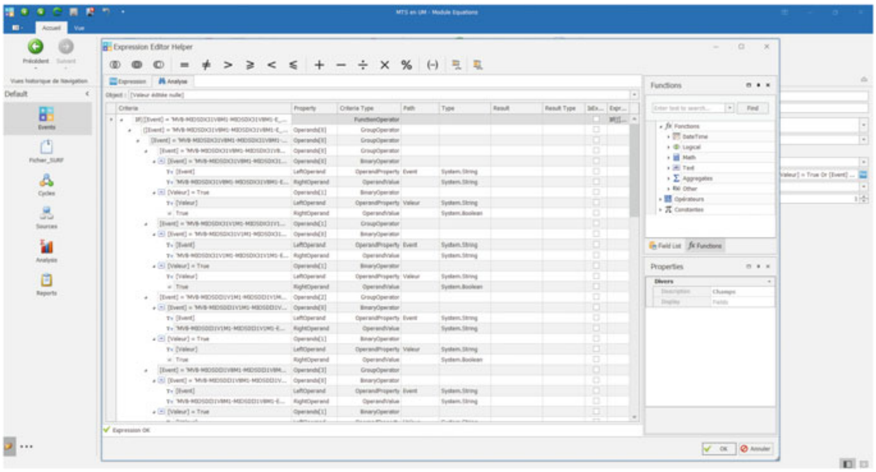


Fig. 4 Screenshot of the TrainSurfer agent rule editor using Surfer data and events

limited but nothing forbids in the near future using them to get a mirrored image in a pure digital world of the TrainSurfer architecture of each train.

From the specification regarding the openness of the architecture, all the recent developments in AI technologies described in the introduction part (vision, clustering, neural network, fuzzy logic…) can be theoretically embedded into specialized TrainSurfer agent. The relationship between these models and the Surfer way of transforming their outputs according to the formatting data (Surfer data) and event (Surfer event) must be respected.

Ultimately, thanks to the TrainSurfer architecture, the interaction with the Human, which can be seen as the tip of the big-data iceberg, is now possible, see Fig. 5. This picture shows a prototype of what is now possible using the

Fig. 5 High-level human-train cooperation prototype based on the TrainSurfer architecture

TrainSurfer architecture. The human interacts directly with the "Train Holon", the latter informing him about the health status of some of its doors and pantograph, calculated by their corresponding SurferAgent.

It is important to note that the TrainSurfer approach does not forbid the use of big data analytics, for example, a Surfer Rule associated to a smart sensor could be (in text form): "`memorize raw data from sensor-1 during 100 last cycles and prepare encrypted zipped file to be sent asynchronously when network bandwidth > threshold`". Off-board studies focusing on long term analyses could be led to identify trends, off line, remotely. This sending is feasible when high connection rates with an infrastructure are available (e.g., in rail station for a train or in airport for a plane). Results from these analytics (e.g., identification of causality between a specific drift in the behaviour of a component onto the average remaining useful life of the equipment) can be integrated back, on the fly, as a new expert knowledge into the corresponding TrainSurfer agent. With such an approach, the advantages of big data analytics are exploited and their drawbacks controlled.

4 Synthesis and Lesson Learnt

Up to now, the TrainSurfer architecture is not fully deployed, but first successful implementation steps have been reached and some trains in use currently embed basic TrainSurfer functionalities. With a full deployment of the TrainSurfer Architecture yet to come, the industrial partner estimates the improvement of the availability of a fleet around 2%. This percentage seems to be low, but applied on an average of 100 trains per fleet, it corresponds to nearly two trains, which means:

On the one side, that the train operator could reduce its fleet from 100 to 98 trains, which translates a potential saving of several millions euros with the same service rate,

On the other side, that the train operator could use this extra-availability to transport more people a year, potentially millions of, in highly saturated urban networks (see introduction part).

Concerning the *maintainability*, it is expected that the TrainSurfer architecture will facilitate the organization of opportunistic preventive maintenance operations when the fleet is less used by the operator (e.g., during nights or middays). An impact of *reliability* is also expected, since the searching for root fault causes is facilitated. More, corresponding knowledge could be used to improve the reliability of next generations of equipment and systems. *Legal investigations* are also facilitated to discriminate responsibilities after a fault (misuse by the train operator versus design error…), which is an important aspect because of the high amount of corresponding financial penalties. Since TrainSurfer Agents memorize the complete history of an equipment or its components, an impact on *sustainability* is also expected, and especially to decide on second-life reuse or recycling processes, including pollutant and fluids.

It is important to note that, thanks to the "Surfer way" the existing train architecture is impacted but not changed. Only new SurferAgents are connected to existing networks and are able to collect the data, process with operator rules where it is necessary, and communicate to increase the efficiency of the day-to-day decisions taken by the fleet operator to optimize his fleet performance. Intermediate value and less important information are erased. The information chain from the sensors and human feelings up to the maintenance is "taylorized". The information flows through enchained blocks with added value at each step: huge quantities of raw, various, inconsistent, un-contextualized, unreliable small data are transformed into accurate digital knowledge understandable by human as well as by systems themselves, which can be summarized as: "from big data to high-accuracy knowledge".

The appropriation of the "Surfer way" and the TrainSurfer architecture by engineers of Bombardier is effective. Two examples illustrate this: first, the TrainSurfer architecture is now integrated as a differentiating offer in response to international bids. Second, as a side effect, engineers from other business units ask the TrainSurfer team to develop new rules, indicators, data and event observers that were not conceived in the initial design phase of the concerned fleet or trains and that solve issues they were facing for several years. Answering this need consumes really less time compared to a new design from scratch because of the intrinsic *modularity* of the holonic approach of the Surfer Architecture.

As a final comment, the authors would like to mention that the closest contributions in the literature to their own work are, from their point of view, the works presented in [26] and in [1].

5 Conclusion and Future Works

The aim of this chapter was to present the "Surfer way", which is a new way of thinking the development of data management architectures. Following these principles, an architecture named Surfer Architecture, aiming to improve the availability and the maintainability of a fleet of transportation systems, was described. An application in the field of train transportation, called "TrainSurfer" was presented. First experimentations are encouraging and the industrial partner has completely appropriated this approach. The main current limit of our contribution is its applicability only on train transportation, which makes it impossible for the moment to validate the real genericity of the Surfer Architecture or the one of its deployment method facing other application fields. Therefore, ongoing works concern the application of the Surfer Architecture to other kinds of transportation fleets.

Acknowledgements This work is done within the context of a joint research Lab, "Surferlab" (http://www.surferlab.fr/en/home), founded by Bombardier Transport, Prosyst and the University of Valenciennes and Hainaut-Cambrésis. SurferLab is scientifically supported by the CNRS and is partially funded by ERDF (European Regional Development Fund). The authors would like to thank the CNRS, the European Union and the Hauts-de-France region for their support. The authors would like to acknowledge Adam Fadil for the development of the TrainAgent rule editor presented Fig. 4 and Quentin Grandin for the development of the Human-Train Holon Interface presented in Fig. 5.

References

1. Tambe, S., Bayoumi, A.-M.E., Cao, A., McCaslin, R., Edwards, T.: An Extensible CBM Architecture for Naval Fleet Maintenance Using Open Standards. Presented at the Intelligent Ship Symposium, Boston, USA (2015)
2. Trentesaux, D., Knothe, T., Branger, G., Fischer, K.: Planning and control of maintenance, repair and overhaul operations of a fleet of complex transportation systems: a cyber-physical system approach. In: Service Orientation in Holonic and Multi-agent Manufacturing, Studies in Computational Intelligence, vol. 694, pp. 175–186. Springer International Publishing (2015)
3. Bukowski, L.: System of systems dependability—theoretical models and applications examples. Reliab. Eng. Syst. Saf. **151**, 76–92 (2016)
4. Lin, X.S., Li, B.W., Yang, X.Y.: Engine components fault diagnosis using an improved method of deep belief networks. In: 2016 7th International Conference on Mechanical and Aerospace Engineering (ICMAE), pp. 454–459 (2016)
5. Cusano, C., Napoletano, P.: Visual recognition of aircraft mechanical parts for smart maintenance. Comput. Ind. **86**, 26–33 (2017)
6. Elattar, H.M., Elminir, H.K., Riad, A.M.: Prognostics: a literature review. Complex Intell. Syst. **2**, 125–154 (2016)
7. Baptista, M., de Medeiros, I.P., Malere, J.P., Nascimento Jr., C., Prendinger, H., Henriques, E. M.P.: Comparative case study of life usage and data-driven prognostics techniques using aircraft fault messages. Comput. Ind. **86**, 1–14 (2017)
8. Yokoyama, A.: Innovative changes for maintenance of railway by using ICT–to achieve "smart maintenance". Procedia CIRP **38**, 24–29 (2015)
9. Thaduri, A., Galar, D., Kumar, U.: Railway assets: a potential domain for big data analytics. Procedia Comput. Sci. **53**, 457–467 (2015)
10. Meng, H., Xu, H., Tan, Q.: Fault diagnosis based on dynamic SVM. In: 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), pp. 966–970 (2016)

11. Xie, H., Shi, J., Lu, W., Cui, W.: Dynamic Bayesian networks in electronic equipment health diagnosis. In: 2016 Prognostics and System Health Management Conference (PHM-Chengdu), pp. 1–6 (2016)
12. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**, 2787–2805 (2010)
13. Lee, T., Tso, M.: A universal sensor data platform modelled for realtime asset condition surveillance and big data analytics for railway systems: developing a smart railway mastermind for the betterment of reliability, availability, maintainability and safety of railway systems and passenger service. In: 2016 IEEE Sensors, pp. 1–3 (2016)
14. Katsouros, V., Koulamas, C., Fournaris, A.P., Emmanouilidis, C.: Embedded event detection for self-aware and safer assets. IFAC-PapersOnLine **48**, 802–807 (2015)
15. Lee, E.A.: Cyber Physical Systems: Design Challenges. EECS Department, University of California, Berkeley (2008)
16. Giret, A., Botti, V.: Holons and agents. J. Intell. Manuf. **15**, 645–659 (2004)
17. Trentesaux, D., Branger, G.: Data management architectures for the improvement of the availability and maintainability of a fleet of complex transportation systems: a state-of-the-art review. In: Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in Computational Intelligence, vol. 762, Springer (2018)
18. Trentesaux, D.: Distributed control of production systems. Eng. Appl. Artif. Intell. **22**, 971–978 (2009)
19. Le Mortellec, A., Clarhaut, J., Sallez, Y., Berger, T., Trentesaux, D.: Embedded holonic fault diagnosis of complex transportation systems. Eng. Appl. Artif. Intell. **26**, 227–240 (2013)
20. Fadil, A., Clarhaut, J., Branger, G., Trentesaux, D.: Smart condition based maintenance (S-CBM) for a fleet of mobile entities. Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in Computational Intelligence, vol. 694, pp. 115–123. Springer, Cham (2017)
21. Mbuli, J., Trentesaux, D., Clarhaut, J., Branger, G.: Decision support in condition-based maintenance of a fleet of cyber-physical systems: a fuzzy logic approach. Presented at the IEEE Intelligent Systems Conference, London (2017)
22. Trentesaux, D., Millot, P.: A human-centered design to break the myth of the « Magic Human » in intelligent manufacturing systems. In: Service Orientation in Holonic and Multi-Agent Manufacturing, Studies in Computational Intelligence, vol. 640, pp. 103–114. Springer (2016)
23. Sallez, Y., Berger, T., Deneux, D., Trentesaux, D.: The lifecycle of active and intelligent products: the augmentation concept. Int. J. Comput. Integr. Manuf. **23**, 905–924 (2010)
24. Trentesaux, D., Rault, R.: Designing ethical cyber-physical industrial systems. In: IFAC-PapersOnLine, pp. 14934–14939 (2017)
25. Trentesaux, D., Rault, R.: Ethical behaviour of autonomous non-military cyber-physical systems. In: the XIX International Conference on Complex Systems: Control and Modeling Problems, Samara, Russia, September (2017)
26. Lee, J., Bagheri, B.: Cyber-physical systems in future maintenance. In: Amadi-Echendu, J., Hoohlo, C., Mathew, J. (eds.) 9th WCEAM Research Papers, pp. 299–305. Springer International Publishing (2015)

Situation Awareness in Product Lifecycle Information Systems

William Derigent and André Thomas

Abstract Years ago, advances in electronics led to electronic components such as RFID, capable to identify a product via a unique identification number. This capacity gave rise to PLIM systems (Product LifeCycle Information Systems), which are information systems ensuring the retrieval of product information wherever and whenever during the product's lifecycle. However, this article shows that such systems are not yet prepared to treat massive data, generated by IoT infrastructure. To solve this issue, situation awareness is a possible answer. In this paper, this notion is detailed and presented via different examples taken from the literature. Yet, the needed tools to support situation awareness in PLIM systems are still lacking, because there are still some scientific and technical challenges to solve. These challenges are detailed in this paper as well.

Keywords Data management · PLIM systems · Situation awareness

1 Impacts of IoT on Data Management in Industrial Systems

Years ago, advances in electronics, computer science and engineering led to the miniaturization of electronic components such as RFID tags at low prices, capable to identify a product via a unique identification number (ID). This unique ID is convenient, since it allows to track a specific product all along its lifecycle, and to link this physical product to a set of virtual items (agents, data containers, etc.) composing the virtual representation of the product.

W. Derigent (✉) · A. Thomas
CRAN, Research Centre for Automatic Control of Nancy, CNRS, UMR 7039,
Campus Sciences, BP 70239, 54506 Vandoeuvre-lès-Nancy Cedex, France
e-mail: william.derigent@univ-lorraine.fr

A. Thomas
e-mail: andre.thomas@univ-lorraine.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_10

In the framework of Holonic Manufacturing Systems (HMS), this link is clearly relevant to implement the concept of Holon, introduced by Koestler [1]. A Holon is an entity composed of a physical part and a virtual part, both associated via a link, referred to as a 'transmutation link' in the rest of the paper (the term 'transmutation' is adapted from [2]). The transmutation link has been widely used in Holonic Manufacturing Systems for asset management and production control. In asset management, a product carries physically a tag [3]. This tag allows retrieving product information throughout the supply chain network, thanks to a unique ID stored in the tag. Product information storage can be centralized or distributed. In the centralized model, product information is transmitted and collected to one place, and in the decentralized one, product information can be stored anywhere (even in its own structure [4, 5]). Both architectures have drawbacks or interests but, most of the time, product information management is done while using a centralized model [6, 7], where the product only owns an unique identification (as EPC Code [8] or ID@URI [9]). Given this unique identification, product information could be requested from a distant database. This mode of management gave rise to PLIM systems (Product LifeCycle Information Systems) [10], which are information systems ensuring the availability of product information wherever and whenever during the product lifecycle. Kiritsis [11] also underlines the fact that information generated by the product item during its lifecycle should be returned to the PLIM systems, in order to: (1) observe the current state of the corresponding product item, and (2) generate new knowledge or modify the existing one from an analysis of these data.

Three well known industrial and academic information management system architectures are based on an object centric paradigm for managing product information at the individual product instance level [12]:

- EPC information system with its standard interfaces for collecting and accessing product-related data [13];
- The approach taken by the DIALOG information system using its ID@URI approach and further developed within the PROMISE project,
- World Wide Article Information (WWAI) approach using a peer-to-peer (P2P) lookup method to access and store data in backend systems.

DIALOG information and EPC network architectures are even standardized ([14, 15]). In parallel, some works try to deal with decentralized architectures, where products can store information on their tags, embedded directly in the product structure. In such situation, some other problems arise like data synchronization between all sources of product information [16]. Both standards (GS1 EPCIS and O-MI/O-MF) provide description of communication interfaces and data structuration to develop a PLIM system.

In the framework of product information sharing, numerous efforts have also been done to propose product ontologies that could describe product information for product data exchanges all over the product's lifecycle [17]. In [18] a review on research works related to PLIM systems is made, which concludes that, apart from the aggregation/disaggregation of information, existing platforms and standards provide efficient ways to share product data.

However, the advent of the IoT slightly changes the perspectives of PLIM usages: the amount of data generated by products becomes massive and can lead to the phenomenon of 'data overload'. Moreover, this data can be used to generate new knowledge. The next section details this phenomenon and provides some examples of data management systems, proposed by the literature, trying to deal with it. This short review shows that each system has capabilities related to *situation awareness*, i.e. the correct perception, comprehension and projection of events occurring in/around the system of interest. These capabilities are critical to cope with Big Data and Knowledge Management in IoT environments. However, PLIM systems do not have the necessary layers to propose generic services dedicated to *situation awareness*, as underlined by the third section. These systems should also evolve from *information management* to *knowledge management*, by integrating the necessary tools to formalize, discover and compare knowledge, in different forms (rules, analytical models, discrete models, etc.). Once that said, the last section tries to identify the remaining scientific challenges to solve in order to design generic architectures for *Product Lifecycle Knowledge Management Systems (PLKMS).*

2 The Situation Awareness in PLIM Systems

Recently, the "Internet of Things" (IoT) was introduced, and gave the possibility to build 'things', i.e. physical objects with cheap and efficient electronic devices as RFID tags but also sensor nodes, able to retrieve/send data from/to the internet. The IoT technologies help to construct a global infrastructure enabling advanced services, by interconnecting these things based on existing and evolving interoperable information and communication technologies [19]. The thing can also be called "connected object" or "communicating object". This link between the real and virtual world is particularly interesting for applications where product status monitoring (e.g. product temperature, localization, speed, orientation, etc.) is considered.

IoT thus brings an easier way to link products to their data. Moreover, the diversity and amount of data collected from IoT is far more important than it was with proprietary solutions, due to the wide adoption of connected objects for industrial or domestic applications [20]. IoT thus generates much and more diverse information than ever before, and this is believed to be an opportunity for manufacturers, since data can serve to better plans, optimize and maintain their processes (Fig. 1). However, uncontrolled IoT can also generate large amounts of unstructured and unnecessary data, leading to 'data deluge' or 'data overload', i.e. bytes stored mechanically, without any real aim or usage afterwards.

In the maintenance domain, some authors are already facing this problem. In [21], a cyber-physical framework is proposed for self-aware and self-maintenance

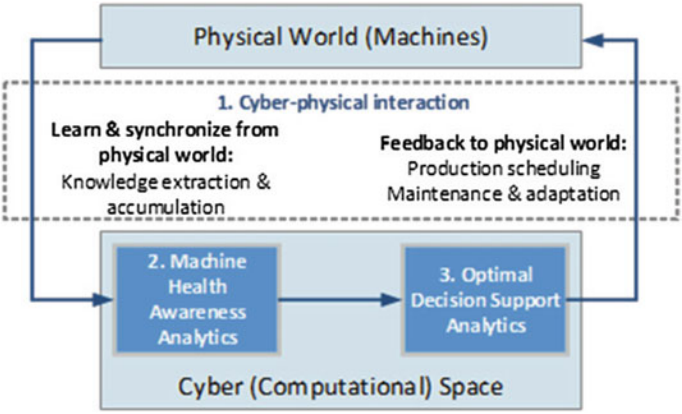


Fig. 1 Proposed cyber-physical framework [21]

machines, that extracts meaningful information data collected from the shop floor machines, and further performs more intelligent decision- making tasks.

Figure 1 represents a simplified vision of the framework proposed in [21], adapted to machine maintenance. It depicts the two worlds (Physical and Virtual—or Cyber in this figure), and the transmutation links between both (referred as Cyber-physical interactions). In the cyber space, two steps are executed, namely *Machine Health Awareness Analytics* and *Optimal Decision Support Analytics.* The first one refers to the capacity to be aware of the health of a machine from a maintenance point of view (i.e. diagnosis and prognosis), and the second to the capacity to take optimal maintenance decisions from the machine health awareness.

Solving the big data challenge can also be made by using decentralized data management architecture as presented in [22], where a framework dedicated to the monitoring of train door was introduced. In this work, a holonic model for the door monitoring was proposed, in which the door holon was responsible of three actions: *Prediction*, *Context Analysis* and *Diagnosis* (see Fig. 2). Each door holon thus treats the data related to its particular physical item, which dramatically decreases the complexity of data processing.

It can be noticed that both works do need to introduce a process constructing an accurate vision of the current system (*Machine Health Awareness Analytics*) for [21], (*Prediction, Context Analysis, Diagnosis*) for [22]). This vision then confers to the decisional entity (agent or human) a better awareness of the system situation. Some questions then arise: *Is such a process always needed? Does a generic process exist?*

In fact, answers can come from works on *Situation Awareness*, defined by [23] as ‘the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future’. This term, mainly used by the military domain until the 80’s, is now widely spread in different other domains like air traffic control, nuclear power plant

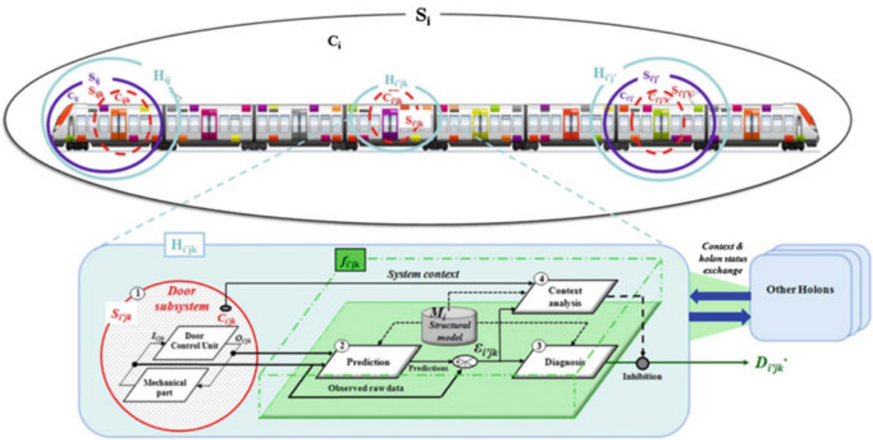


Fig. 2 Model of Holonic door monitoring

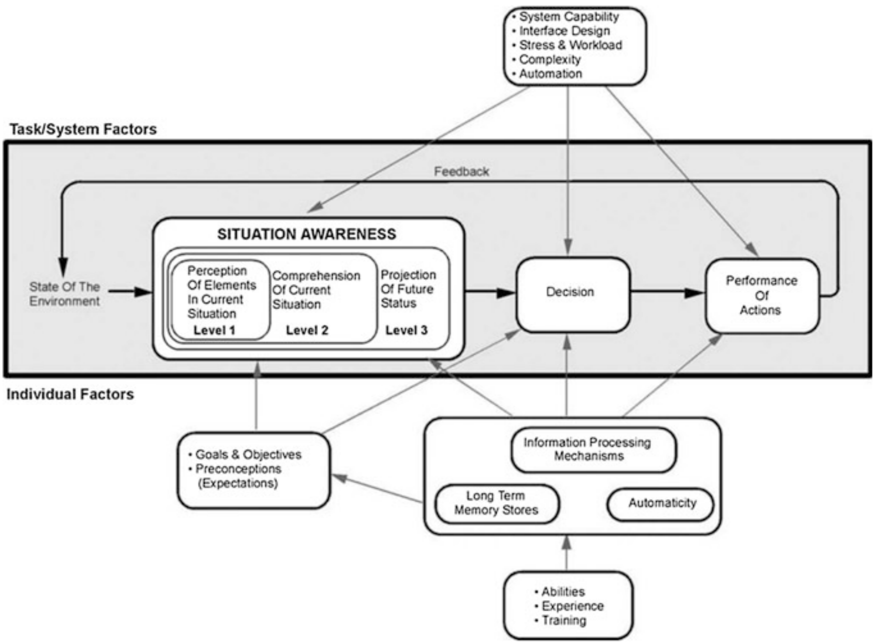


Fig. 3 Endsley's situation awareness model [23]

operation, vehicle operation, etc. Figure 3 shows Endsley's situation awareness model. In this model, the situation awareness is a 3-step process composed of the following elements: (1) perception of elements in current situation (level 1), (2) comprehension of current situation (level 2), and (3) projection of future status (level 3).

3 The Necessary Integration of Situation Awareness into PLIM Systems

Let's now consider identical machines in different companies, but remotely monitored by the same maintenance company. In such situation, a PLIM system may be used for product data collection and retrieval (level 1 of situation awareness). In current implementations of PLIM, a product-centric approach is often used, meaning that each individual item is linked to its virtual representation, i.e. a software agent managing the data of the real item. Basically, it means that there is a bijection between each real machine and each monitoring agent belonging to the PLIM system. PLIM systems proved their efficiency in such situation (the PROMISE project [7], the AERO-ID project [24] report many use cases in that sense).

However, when it comes to situation awareness, knowledge management (extraction, update, accumulation, retrieval and use) is also required, to understand the current machine situation (level 2 of situation awareness), Fig. 4. Moreover projection capabilities are also needed to compute possible future status of the considered machine (level 3 of situation awareness). Current PLIM systems do not support these types of functionalities. Indeed, in these systems each software agent should be able to clearly know the situation of its physical item (current item state

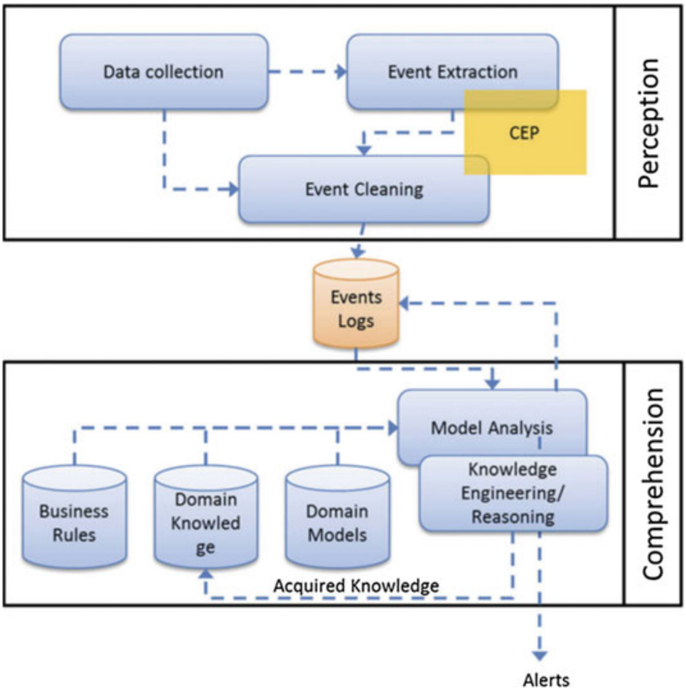


Fig. 4 Details of the situational awareness module

and current environment state), at least to notify a human operator or to take decisions based on it.

To our knowledge, in the manufacturing and logistics domain, the concept of situation awareness has only been addressed by [25]. The authors present an IoT-based situational awareness framework for real-time project management, composed of 2 layers, namely the *IoT* and *Situational awareness* layers. The first one is responsible for collecting events from shop-floor to provide real-time feed of data into the system. The second one is divided into 2 sub-layers, perception and comprehension (level 1 and 2 of situation awareness), that are respectively responsible for the collection and processing of data, and the generation of alerts, based on model analysis, knowledge engineering and reasoning tools. This layer obviously needs contextual information, given by dedicated domain models, domain knowledge and business rules. Figure 4 gives details of the situational awareness layer. This framework is interesting by its genericity: although applied initially for project management in the construction industry, it seems to be suitable for maintenance or manufacturing operations.

As a result, the existing PLIM systems should be completed by a generic situational awareness layer, capable to consider and link multiple types of models, in order to be applied to several domains (manufacturing, construction, energy or even maintenance). Example of models can be manufacturing cell simulation models, thermal models, rules, etc. In such system, each physical item would then be linked to a virtual entity that can mimic its behaviour. Each virtual item can then be considered as a 'digital twin' of the physical item, encapsulating data but also knowledge related to the physical item. Adding such a layer would allow evolving from Product Lifecycle Information Management Systems to *Product Lifecycle Knowledge Management Systems*.

4 Remaining Technical and Scientific Challenges

As stated before, PLIM systems must evolve towards PLKM systems. In this section, the remaining technical and scientific challenges are detailed and grouped according to the steps of the *Situation awareness* process detailed Fig. 3, i.e. *Perception, Comprehension* and *Projection*:

- *Perception*: The perception problems relate to the collection and fusion of multiple imprecise sources of data, with different degrees of confidence. The diversity of data is highly important in such environment: data can be originating from 'things' and collected thanks to IoT standardized protocols, from other internet sources (online forecast, communication with distant Enterprise Information Systems, etc.) or even from user interactions. In [26], the authors advocates that improved situation awareness requires high-level data representations. They thus emphasize the need to develop semantic fusion based IoT architecture, coupling traditional data fusion models with a semantic layer.

- *Comprehension*: Here also [26] underlines that an improved situation awareness demands human-like intelligence and reasoning capabilities. PLKM Systems must then be adapted to do so. As shown in Fig. 4, comprehension also requires to perform analysis based on models. These models must be managed by the PLKM Systems, and should be able to be connected to each other when performing situation awareness. This network of models could then be used to evaluate the system situation from the perceived data. This implies new data management architectures as well as interoperability issues between these models. The initial system models can be obtained either via human expertise or generated automatically [27]. Once generated, the model validity should be checked all along the system lifecycle and the model has to be updated if obsolete. The PLKM Systems should at least provide some mechanisms to evaluate the pertinence of a model during its use. The comprehension is also related to the structuration of the data management architecture, as shown in Fig. 2, where using a holonic model facilitates the diagnosis process. If one considers true the assumption that the holonic model is the most adapted for industrial applications, the PLKM should therefore include some mechanisms to represent the aggregation/disaggregation mechanisms of the holonic theory. For example, the holon 'car' can be also viewed as the aggregation of holons 'chassis', 'wheels' and 'engine', not only the different data but also the different behaviours. The VSM based product-driven architecture proposed in [28] and illustrated in Fig. 5 is probably a good candidate for data modelling in PLKM systems.

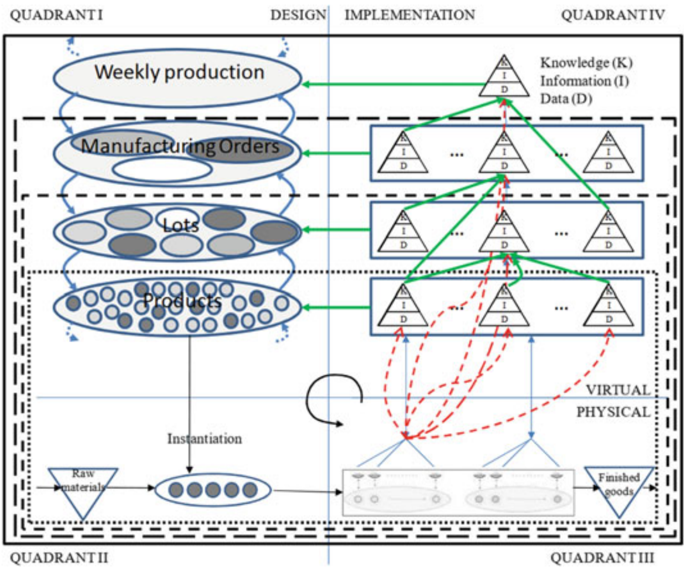


Fig. 5 VSM based product driven control system

- *Projection*: The projection of a complete system is a hard task, prone to numerical complexity and combinatorial explosion. It refers to the extrapolation of the current situation forwarded in time to determine how it will affect future states of the operational environment. Several models are available, depending on the domain of interest. However, a PLKM system should provide a generic infrastructure to enable and evaluate the different projections, based on projection models available in the system. This notion is hardly addressed in the industrial Internet of Things.

5 Conclusions

In the area of IoT, there is a strong opportunity to evolve from PLIM systems to PLKM systems that has been underlined in this paper. Some approaches dealing with the IoT demonstrate that the use of additional mechanisms allows tackling data overload as well as to create new knowledge used for product monitoring, as an example. Compared to PLIM, PLKM systems must have additional generic functionalities, needed to implement a process of situation awareness, as defined in [23]. However, some research work is still required because theories, methods and tools to perform the actions of perception, comprehension and projection in diverse, distributed and constrained environments are not completely available.

References

1. A. Koestler: The ghost in the machine, Hutchinson (1967)
2. Le Gallou, F., Bouchon-Meunier, B.: Systémique: théorie et applications (1992)
3. Meyer, G.G., Främling, K., Holmström, J.: Intelligent products: a survey. Comput. Ind. **60**(3), 137–148 (2009)
4. Kubler, S., Derigent, W., Thomas, A., Rondeau, É.: Embedding data on 'communicating materials' from context-sensitive information analysis. J. Intell. Manuf. **25**(5), 1053–1064 (2014)
5. Mekki, K., Derigent, W., Zouinkhi, A., Rondeau, E., Thomas, A., Abdelkrim, M.N.: Non-localized and localized data storage in large-scale communicating materials: probabilistic and hop-counter approaches. In: Computer Standards and Interfaces (2015)
6. Främling, K., Harrison, M., Brusey, J., Petrow, J.: Requirements on unique identifiers for managing product lifecycle information: comparison of alternative approaches. Int. J. Comput. Integr. Manuf. **20**(7), 715–726 (2007)
7. PROMISE: Product lifecycle management and information tracking using embedded systems. http://www.promise.no. Accessed June 2017
8. E. P. C. Global: EPC global Architecture Framework. http://www.epcglobalinc.org/standards/architecture (2009). Accessed 12 Nov 2009
9. Kärkkäinen, M., Ala-Risku, T., Främling, K.: The product centric approach: a solution to supply network information management problems? Comput. Ind. **52**(2), 147–159 (2003)
10. Främling, K., Holmström, J., Loukkola, J., Nyman, J., Kaustell, A.: Sustainable PLM through intelligent products. Eng. Appl. Artif. Intell. **26**(2), 789–799 (2013)

11. Kiritsis, D.: Closed-loop PLM for intelligent products in the era of the Internet of Things. Comput. Des. **43**(5), 479–501 (2011)
12. Brintrup, A., McFarlane, D., Ranasinghe, D., Sanchez Lopez, T., Owens, K.: Will intelligent assets take off? Toward self-serving aircraft. IEEE Intell. Syst. **26**(3), 66–75 (2011)
13. Ranasinghe, D.C., Harrison, M., Cole, P.H.: Networked RFID Systems and Anti-counterfeiting: Raising Barriers to Product Authentication. Springer (2007)
14. The Open Group O-MI: Open Messaging Interface, An Open Group Internet of Things (IoT) Standard, Reference C14B, US ISBN 1-937218-60-7 (2014)
15. EPCgloabl: http://www.epcglobalus.org/. Accessed June 2017
16. Suzuki, S., Harrison, M.: Data synchronization specification, Aerospace-ID program report (2006)
17. El Kadiri, S., Kiritsis, D.: Ontologies in the context of product lifecycle management: state of the art literature review. Int. J. Prod. Res. **53**(18), 5657–5668 (2015)
18. Derigent, W., Thomas, A.: End-of-life information sharing for a circular economy: existing literature and research opportunities. In: Service Orientation in Holonic and Multi-Agent Manufacturing, pp. 41–50. Springer (2016)
19. Harvard Business Review: Internet of things: science fiction or business fact? Harv. Bus. Rev. Anal. 8 (2014)
20. Xu, L., He, W., Li, S.: Internet of things in industries: a survey. IEEE Trans. Ind. Inform. **99** (2014)
21. Lee, J., Ardakani, H.D., Yang, S., Bagheri, B.: Industrial big data analytics and cyber-physical systems for future maintenance and service innovation. Procedia CIRP **38**, 3–7 (2015)
22. Le Mortellec, A., Clarhaut, J., Sallez, Y., Berger, T., Trentesaux, D.: Embedded holonic fault diagnosis of complex transportation systems. Eng. Appl. Artif. Intell. **26**(1), 227–240 (2013)
23. Endsley, M.R.: Theoretical underpinnings of situation awareness: a critical review. In: Situation Awareness Analysis and Measurement, pp. 3–32 (2000)
24. Harrison, M., Parlikad, A.K.: Lifecycle ID and Lifecycle Data Management (2006)
25. Ghimire, S., Luis-Ferreira, F., Nodehi, T., Jardim-Goncalves, R.: IoT based situational awareness framework for real-time project management. Int. J. Comput. Integr. Manuf. **30**(1), 1–10 (2016)
26. Singh, D., Tripathi, G., Jara, A.J.: A survey of Internet-of-Things: Future vision, architecture, challenges and services. In: 2014 IEEE World Forum Internet Things, WF-IoT 2014, pp. 287–292 (2014)
27. Véjar, A., Charpentier, P.: Generation of an adaptive simulation driven by product trajectories. J. Intell. Manuf. **23**(6), 2667–2679 (2012)
28. Thomas, P., Thomas, A.: An Approach to Data Mining for Product-Driven Systems. Studies in Computational Intelligence, vol. 472, pp. 181–194. Springer (2013)

Part III
Cyber-Physical Production Systems

Proportional Reliability of Agent-Oriented Software Engineering for the Application of Cyber Physical Production Systems

Luis Alberto Cruz Salazar and Hang Li

Abstract *Cyber Physical Production System* (CPPS) is one of the most significant concepts of the Industry 4.0, which has attracted spread attention from both academic community and industry. It is widely accepted that the *Industrial Internet of Things* (I^2oT) and advanced methodologies for manufacturing systems are essential factors to achieve CPPS. Among the methodologies, a class of agent-oriented methodologies is considered prevailing in the manufacturing scenarios. Thus, some promising methodologies are proposed based on agent-oriented architecture. However, most of these methodologies need to be further evaluated. This paper reviews the evolution of the software architecture and comes up with the requirements of the validation of these methodologies. According to the proposed requirements, the latest methodologies are analyzed and the future research roadmap is proposed.

Keywords Agent-oriented software engineering · CPPS · Requirements specification · Methodology evaluation · Industry 4.0

1 Introduction

The software paradigms for production control have evolved from a centralized architecture to a decentralized or semi-heterarchical one [1]. The evolution is mainly driven by the new trends in information and communication technology (ICT) such as Industrial Internet of Things, agent technology and so forth. Agents and multi-agent systems (MAS) have drawn enormous research interests from the computer science community and industry. Agent-oriented technology is targeting

L. A. Cruz Salazar (✉) · H. Li
Institute of Automation and Information Systems, Technical University of Munich, Garching near Munich 85748, Germany
e-mail: luis.cruz@tum.de

H. Li
e-mail: hang.li@tum.de

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_11

the software development on concurrency, consistency, negotiation between agents and environment with the deployment of intelligent entities—agents [1]. Because of its efficiency on design, analyses, and construction for complex software systems, it is widely accepted as one pillar of Cyber Physical Production System, which is a crucial technique for Industry 4.0. Another pillar for CPPS is Industrial Internet of Things. However, the industrial network can be improved with consistent agent-oriented methodologies. Some promising agent-oriented methodologies are proposed at least five years ago. The resent research efforts have been put into designing smart systems by themselves, but little or even no attention was given to the development of comprehensive "smart, safe and sustainable" systems [2]. McFarlane et al. explore the intelligence in the evolving industrial control paradigm.

This paper is based on their works; it tries to classify the requirements to validate these methods for further simulation and implementation in industrial automation based on MAS. The rest of the paper is organized as follows. Section 2 introduces the evolution of the software architecture for manufacturing systems. The fundamental pillars of Industry 4.0 are discussed in Sect. 3. The requirements for agent-oriented methodologies are proposed and analysed in Sect. 4. Section 5 concludes the paper and comes up with the future research roadmap.

2 Evolution of Software Architectures for Manufacturing Systems

This section reviews the evolution of system structure for manufacturing system on the control architecture. System architecture falls into four categories according to the decision-making mechanism. The most simplified structure for a manufacturing system relies on an intuitive and *centralized model*. In the model, an element is assigned to a central control unit with the highest priority, which governs and monitors the execution of sub-execution systems. The centralized model falls into two categories according to the operating process of the sub-execution systems (in sequence or parallel). The call–return sub-model is an outright top-down scheme where a process starts at the top of the hierarchy and, through subroutine calls, passes to lower levels in the decision. This model is only suitable for sequential systems. The manager sub-model is applied in parallel systems. In the model, one component is designated as the manager responsible for the entire system process. The comparison of these two models is displayed in Fig. 1.

The second type is the *fully hierarchical system*. It is the upgraded level of the centralized system. There is a control hub between the central control unit and the lower execution unit. The third kind of decision-making mechanism is the *heterarchical model*. In lower level, components operate on local process to achieve global goals. Compared with Fig. 1a, a central decision entity does not exist in the model. Heterarchical systems are intricately linked to the idea of self-organization

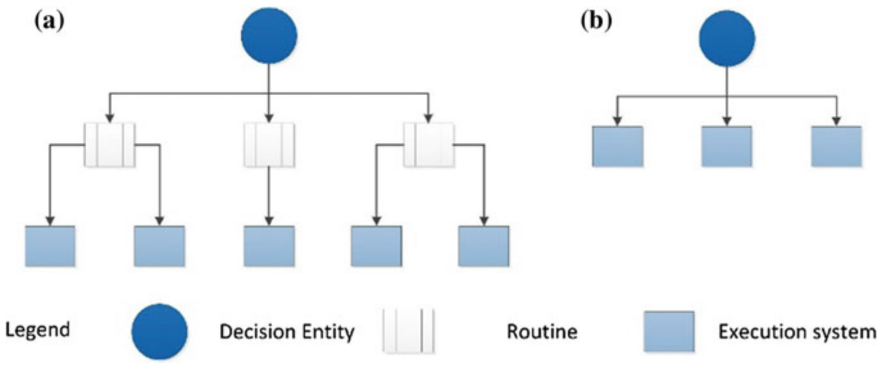


Fig. 1 The comparison of call-return sub-model and manager sub-model

because local interactions between sub systems of a decentralized scheme achieve global goals in their coordination without a central control unit. The last type of system structure is the *semi-heterarchical system*. It distributes global targets of a system into a set of individual targets for sub systems. The concept of semi-heterarchical system improves reliability and reduces installation costs by controllers on the corresponding plant, while remote control and diagnose is achievable in the model. In fact, it is possible to exploit commonly used proportional reliability measures for quantitative data to facilitate identifying constant and "real" info characteristics in practical designs.

A control structure can also fall into four typological classes, classified from Class 0 to Class III, according to Christensen [3], Trentesaux [4] and Cruz Salazar [5], see Fig. 2.

From Fig. 2, one can notice two paths from a centralized system to heterarchical systems. Some production techniques which were given new names such as "lean" or "slim" lead to a turning point in the whole management philosophies architectures. These prevailing system architectures are MAS and holonic systems (HMS). Based on [3], HMS is a holarchy (holons/parts groups) which integrates the entire range of manufacturing activities. UPH form is a promising paradigm from HMS, Fig. 2 [5].

In this paper, only the software for CPPS is concerned; holon is beyond the topic of the domain.

Guenther et al. proposed a hierarchical system for scheduling of continuous casters and hot strip mills in the steel industry. The numerical results demonstrate the practicability of the approach under experimental condition [6]. Chung et al. came up with a genetic algorithm approach, with a hierarchical system for perfect maintenance in distributed production scheduling aiming to minimize the makespan of the jobs. Moreover, the approach improves the local searching ability by iteratively solving the jobs allocation problem and scheduling of production [7]. Zhang et al. proposed a hierarchical capacity planning approach for the reconfiguration of kits in the assembly and test semiconductors manufacturing [8]. In agent-oriented

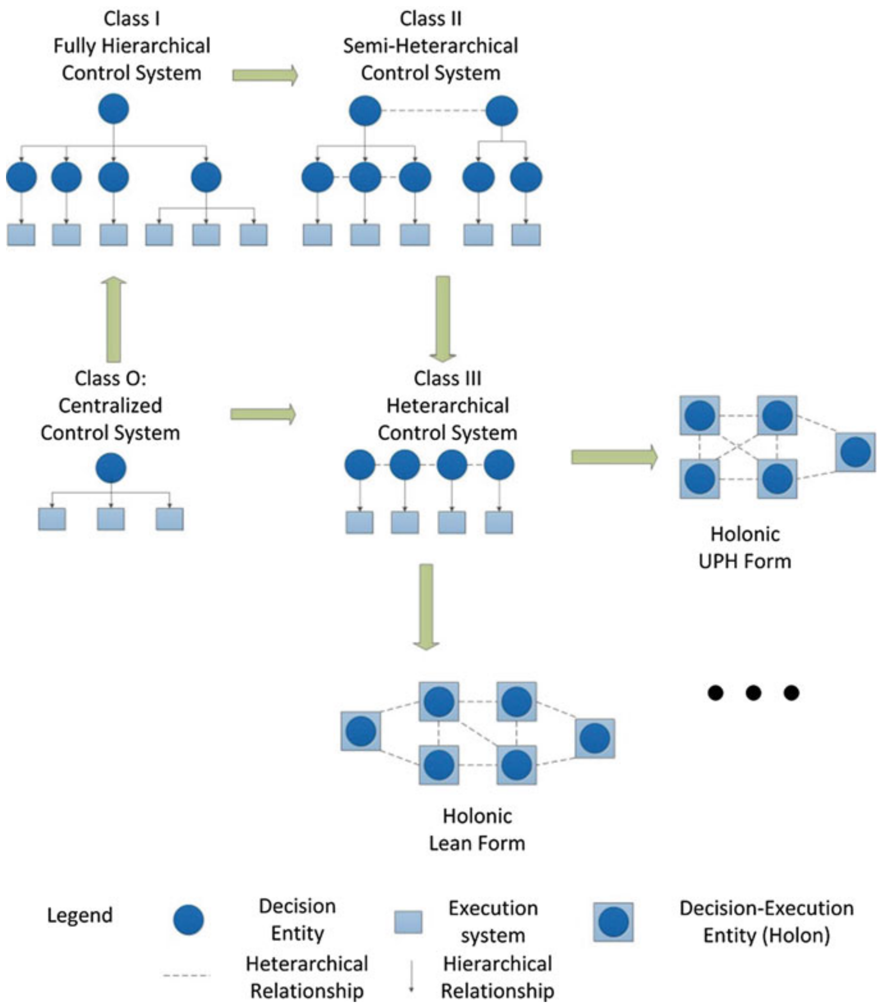


Fig. 2 The evolution of system architecture paradigms

software engineering, the agent is defined as an encapsulated software unit with a defined goal [9]. A multi-agent system is a cybernetic system including multiple intelligent agents which can interact with each other and even with the environment. The MAS has been studied as an independent field right since about 1980. After that, MAS has attracted widespread attention. Since the mid-1990s, general attractiveness in this research domain has grown immensely. This prompt growth has proven that agents are an appropriate software paradigm for massively distributed systems.

3 Industry 4.0 and Cyber Physical Systems for Manufacturing

Technological advances have driven the increase in industrial productivity, since the beginning of the first Industrial Revolution at the end of the eighteenth century. In principle, steam engines fed factories in the nineteenth century. In next century electricity was introduced which led to mass production, and following the industry was automated in the 1970s with the implementation of electronic systems, usually programmable devices. However, in later decades, industrial technological progress has been merely gradually compared with the advancements of Information Technologies and e-commerce. Nowadays, manufacturing systems are in the midst of the Industry 4.0, which is arising from a transformation of the new digital manufacturing era, supported by new technological developments [10].

With I^2oT and CPS, manufacturing systems will be able to collect and analyse data across machines, enabling faster, more flexible and more efficient processes for goods of higher quality. Also, it will modify the professional profile of the work skills of human resource [11, 12]. Ultimately, significant changes in the competitiveness of enterprises, regions, and countries will take place.

Rüßmann et al. reported in [9] that nine techniques are crucial domains for future fabrication of the Industry 4.0, which are displayed in Fig. 3. The potential growth of technology and economy for vendors among the whole supplier chain are discussed in [10, 11].

I^2oT is one relevant pillar of the nine foundations established which includes other characteristics, determinism, reliability, and timing. It is noteworthy that today only some of the sensors and manufacturing machines are networked and made use of the embedded systems. Devices are typically organized in a pyramid vertical

Fig. 3 Nine pillars supporting Industry 4.0 [11]

automation, where sensors, actuators, controllers and other automation components are limited to central control systems.

Diverging and mutually interacting research of manufacturing technologies with computer science, as well as information and I^2oT occasioned in what is called now Cyber Physical Production System. It is frequently noted that CPPS is leading to help the I4.0 revolution, and its concept comprises independent and cooperative components. Subsystems close to the original CPS concept are getting linked with each other with I^2oT on and across all levels of manufacturing systems. It happens from the processes level through mechatronics machines up to fabrication and high-level logistics systems (enterprise management). Another pillar of Industry 4.0 is additive manufacturing which became possible with CPPS development. It is probable to modify or adapt products for specific production changes or client needs. Within Industry 4.0, additive-manufacturing procedures will be extensively used to produce small lots of modified products that offer manufacture advantages (e.g. complex and lightweight designs). Also high performance and decentralization of additive manufacturing systems could decrease transportation and stock systems [9].

3.1 Cyber Physical Production Systems, Roots, Applications and Basic Model

CPS applications include automation, manufacturing systems, medical devices, military systems, assisted living, traffic control and security, process control, energy generation and distribution, HVAC (heating, ventilation and air conditioning), aircraft, instrumentation, water management systems, trains, physical security (access control and surveillance), asset management and distributed robotics (telepresence, telemedicine) [11, 13–16]. As an intellectual challenge, CPS is a concept that is viewed as the intersection, not the union of physics, and cybernetics. It combines engineering models and methods of electronic, electrical, mechanical, environmental, civil, biomedical, chemical, aerospace, and industrial systems with computer science. Several scientists argue that these models and methods are not easily combined and the consequence is that CPS is a new engineering discipline that requires its models and methods [17]. The term "Cyber-Physical Systems" emerged in 2006, and was launched by Helen Gilld of the National Science Foundation of the United States [18]. The related term as "Cyberspace" is proposed by William Gibson, who uses the term in the novel Neuromancer, but the roots of the term CPS are larger and deeper. CPS logic control was indeed a calculation, although carried out with analogue circuits and components. Therefore, cybernetics is the combination of physical processes, calculation, and communication. According to E. Lee in [19] there is a deterministic CPS modeling called Ptides (from programming temporally-integrated distributed embedded systems). This paradigm or metaphor is apt to control systems, and Fig. 4 symbolizes a simple structure of a CPS.

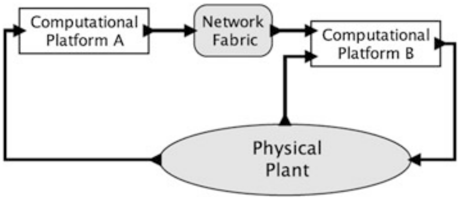


Fig. 4 The structure of a simple Cyber-physical System (CPS) [19]

CPS is constantly related to machine to machine (M2M) model and IoT in Industry 4.0. They reflect the view of obtaining technologies connecting the entire physical world with the world of information. CPS is sometimes confused with the term CPPS, which is intensively linked with the close physical world and its on-going methods for manufacturing. CPPS uses and gives in parallel time networked information accessing and processing services available normally on the internet, e.g. cloud manufacturing [20].

Another term "Cybersecurity" for manufacturing refer to safety of cyberspace, and is therefore only indirectly related to cyber part. A CPPS certainly is involved but not limited to many challenging issues of security and privacy. It is possible to talk about a "theory of CPPS" similar to the "Theory of Linear Systems," since both terms have formal models. Models and methodologies play a central role in all scientific and engineering disciplines. However, one of the main drawbacks of the CPPS is that it combines different disciplines. Unfortunately, the models and methodologies prevailing in these disciplines do not synchronize well, since modelling languages are disjoint and incompatible in semantics.

3.2 General Design Methodology for Cyber-Physical Systems

The design of a CPS, especially sub heterogeneous distributed systems across networks, turns out to be a demanding task. Some design techniques commonly applied in CPS are sophisticated, and they include mathematical models of physical components, computation, and simulations of heterogeneous systems, software synthesis, verification, validation, and testing.

Therefore, it is necessary to identify a series of sequential steps, which are sufficient for the development of simple and complex CPS. For example, based on [19, 21] there is a design methodology for CPS which includes not a set of necessarily sequential steps, but essentially co-dependent ones facilitate the co-evolution of a model for a CPS implementation (see Fig. 5).

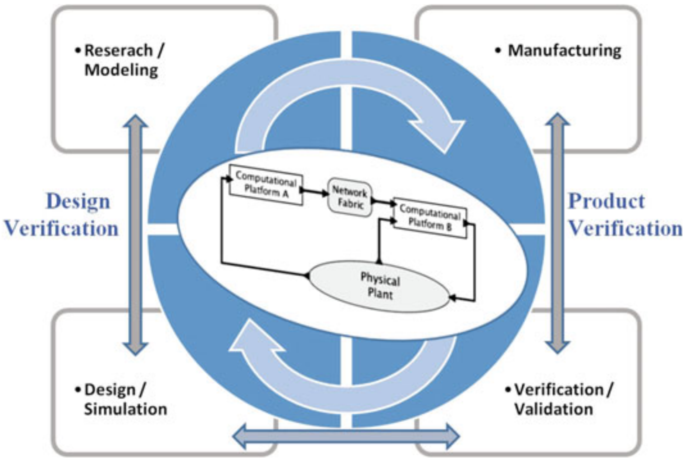


Fig. 5 Design iterative methodology for CPS [21]

3.3 Agent-Oriented Software Engineering Methodologies for CPPS

MAS or Agent-based approaches demonstrates a method to achieve CPPS in nature [22–25]. Agent-Oriented Software Engineering (AOSE) is the core concept of MAS, and several promising AOSE methodologies have been proposed since least ten years [6]. AOSE methodologies highly depend on the MAS requirements, and the present paper tries to classify these requirements.

Taking some relevant software requirements in consideration, it is possible to identify and to determine the different methodologies reported in the dedicated literature of AOSE and extend these procedures into account required by the implementation of CPPS.

Table 1 presents a brief summary of various AOSE methodologies with Concept and Attributes or Terminology (more details can be found in the scientific work reported in [26, 27]).

4 Requirements Terms for Agent-Oriented Methodologies for CPPS

The methodologies of AOSE for the CPPS selected in the previous part are presented with event-driven MAS. A comparison should be introduced based on the following evaluation criteria, which are grouped into four main categories: CPPS Minimal Conditions (CR1); Intelligent Characteristics Attributes (CR2);

Table 1 Main AOSE methodologies selected concepts and terminologies [26, 27]

Main AOSE	Concept	Attributes/Terminology
Gaia	It is composed of a number of autonomous interactive agents that live in an organized society in which each agent plays one or more specific roles	Responsibilities, Permissions, Activities and Protocols
MaSE	It is an organization-based, role-centered process framework including three main elements: a metamodel, method fragments, and guidelines	Capabilities, Requirements, Plan Selection Algorithm, Execution Component Execution Algorithms, Control Component Execution Algorithms, State transitions and Behaviours
MESSAGE	It covers analyses and design of MAS which is suitable for application based on the best practices in mainstream software engineering departments	Organization, Role, Resource, Task, Interaction, Protocol, Goal, Information Entity, Message, Analysis Model Views
TROPOS	It is a requirements-driven methodology in the sense which is based on concepts employed during the early stage of requirements analysis	Early requirements, Late requirements, Architectural design, Detailed design
Prometheus	It includes environmental and organizational dimensions in the development process	Artefacts, Dimension Interaction
INGENIAS	It skims the evaluation criteria and facilitates the CPS event-driven multi-agent model development	Organization meta-model, Environment meta-model, Tasks/Goals meta-model, Agent meta-model, Interaction meta-model
SODA	It is addressed to inter-agent problems, and provides explicit abstractions and methods for the engineering of MAS environment or agent societies	Cognition and Agency, Early Reasoning Models
PASSI 2	It consists of five models, which include several distinct phases and it provides PTK (Passi Toolkit) and AgentFactory	System Requirements, Agent Society, Agent Implementation, Code Model and Deployment Model
ASPECS	It provides a step-by-step guide from requirements to code allowing the modelling of a system at different levels of details with a set of refinement methods based on a holonic organizational metamodel	Problem domain, Agency domain, Solution domain
MOBMAS	It makes use of ontologies as a modelling tool and supports the analysis, interaction, internal design, organization and architecture of MAS development	Analysis Activity, MAS Organization Design, Agent Internal Design, Agent Interaction Design, Architecture Design

Table 2 Summary of requirements and items for AOSE to apply CPPS

CPPS Requirement 1: CPPS minimal attributes (CR1)	
R1.1	Independent architecture model which implies that modules are simple to be integrated with independent implementation (open architecture and platform)
R1.2	There is an Open communication protocol for I^2oT which is capable of being easily and quickly switched between open networks
R1.3	All levels of automation for ISA 95 are available depending on the scenarios in which the CPPS will be applied
R1.4	Usability and Adaptability are available to the systems for future products which are called Smart Products
CPPS Requirement 2: Intelligent characteristics (CR2)	
R2.1	Systems have Autonomy, and the updated processing experience is achieved by deducing behaviours of the CPPS based on goals
R2.2	Its communication and ontology express similarly way between similar agents
R2.3	System enables the development of mutually acceptable goals (cooperative skills)
R2.4	System is capable of achieving its assigned goal having initiative skills (Proactivity)
CPPS Requirement 3: Formalized modeling (CR3)	
R3.1	Models for CPPS use standard language and international norms (formalism such as the Unified Modeling Language—UML)
R3.2	System has different degrees of abstraction for applying the model in CPPS
R3.3	CPPS implementation has Integrated Development Environment (IDE) and platforms
CPPS Requirement 4: Systems and Human Integration (CR4)	
R4.1	Systems is open to different systems domain (e.g. electricity supply systems, discrete manufacturing, batch or continuous processes)
R4.2	Hybrid topologies are included to enlarge or downsize the production system
R4.3	Methodologies must provide social norms regarding human factors

Formalized modelling (CR3) and Systems and Human Integration Needs (CR4). Table 2 indicates this requirements summary and its corresponding items.

According to all items of requirements (CR1-CR4) announced, Table 3 indicates these in the first column and, its second column shows the acceptance values considered for main AOSE to apply in CPPS.

4.1 Requirement 1: Proportions of CPPS Minimal Conditions

It is possible to generate coverage proportions of main AOSE for CPPS Minimal Conditions (CR1) from Table 2. Figure 6 indicates portions in the relation of High, Medium and Low levels coverage for each item of requirement CR1.

With data from Table 3 and the analysis of Fig. 6, it is helpful to extract some meaningful conclusions related to the adoption of AOSE for the CPPS Minimal Attributes (CR1). Independence architecture model item is entirely covered

Table 3 Acceptance values considered for items of requirement 1 (2: High, 1: Medium, 0: Low)

CPPS requirements		Main AOSE									
		Gaia	MaSE	MESSAGE	TROPOS	Prometheus	INGENIAS	SODA	PASSI 2	ASPECS	MOBMAS
CR1—CPPS minimal attributes	R1.1	2	2	2	2	2	2	2	2	2	2
	R1.2	2	2	1	0	1	2	0	1	0	0
	R1.3	1	2	0	0	2	1	0	0	0	0
	R1.4	2	2	2	2	1	2	1	1	2	1
CR2—Intelligent characteristics	R2.4	1	2	2	1	1	2	2	2	1	2
	R2.5	2	2	2	1	0	2	2	2	2	1
	R2.6	1	2	2	1	2	1	2	1	1	2
	R2.7	2	0	2	0	0	2	0	0	0	0
CR3—Formalized modeling	R3.1	2	0	2	2	2	2	2	0	2	2
	R3.2	1	0	1	0	0	1	0	1	0	0
	R3.3	1	1	2	1	1	2	0	1	2	0
CR4—Systems and human integration	R4.1	0	2	2	2	2	2	2	1	2	2
	R4.2	2	0	2	2	0	2	0	0	0	0
	R4.3	0	0	1	0	0	1	0	0	2	0

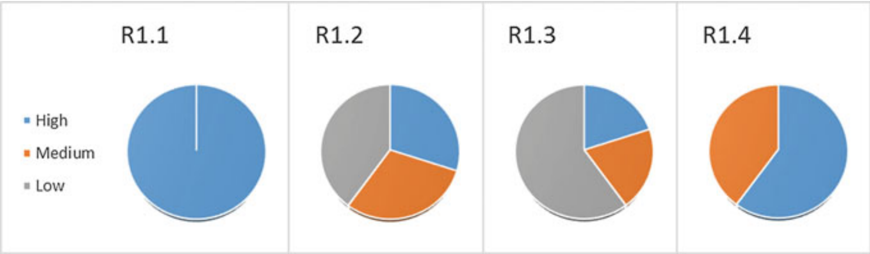


Fig. 6 Proportions of items from requirement 1 (CR1) of AOSE methodologies for CPPS

(R1.1 = 100% High). However, for the same requirement there is a little coverage of the ISA 95 levels for vertical integration automation item (R1.3 = 60% Low). It means AOSE methodologies should help to increment integration of separate system components probably regardless of their location in a plant hierarchy (or global context of ISA level).

4.2 Requirement 2: Proportions of Intelligent Characteristics Attributes

Based on data from Table 3, it is possible to generate coverage proportions of main AOSE for Intelligent Characteristics Attributes of CPPS (CR2). Figure 7 indicates proportions in the relation of High, Medium and Low levels coverage for each item of requirement CR2.

The general distinction in Intelligent Characteristics requirement (CR2) has the adequate coverage. However, Proactivity is not available for many AOSE yet (R2.4 = 70% Low). AOSE methodologies should be improved to allow obtaining more acting initiatives before some event happens if it could affect achieving their assigned goal.

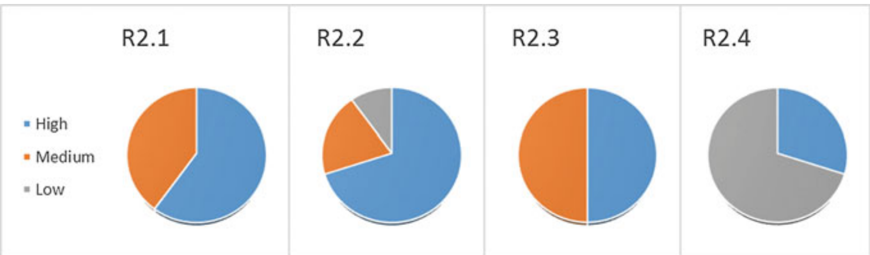


Fig. 7 Proportions of items from requirement 2 (CR2) of AOSE methodologies for CPPS

4.3 Requirement 3: Proportions of Formalized Modeling Terms

It is vital for innovative approaches to abstractions (formalisms) and architectures to enable control, communication and computing integration. For this reason, Formalized Modeling Terms requirement (CR3) could help the rapid design and implementation for CPPS, based on three items (Fig. 8).

With the information from Table 3, it is possible to generate coverage proportions of main AOSE methodologies for Formalize Modeling Terms of CPPS. Figure 8 shows proportions in relation of High, Medium and Low level coverage for each item of requirement CR3.

In general, there is good coverage in the Formalized Modeling Terms requirement (CR3) for AOSE methodologies. Notwithstanding, there is a small level of Abstraction for an Overview item (R3.2 = 60% Low). Then, AOSE will require different formal developing methods depending on the degree and include the integration of bottom-up and top-down schemes.

4.4 Requirement 4: Proportions of Systems and Human Integration

Systems and Human Integration Needs is the last and significant requirement to apply CPPS (CR4), and it is categorized into three items. Through the analysis of Table 3, it is evident to generate coverage proportions of main AOSE for Open Systems to Different Systems domain of CPPS. Figure 9 indicates proportions in the relation of High, Medium and Low levels coverage for each item of requirement CR4.

Observations about items of Systems and Human Integration requirement (CR4) are the least coverage as essential items of this specification are not included yet.

In fact, both Hybrid Topologies (R4.2 = 60% Low) and the Social Norms Considering Human Factors items (R4.3 70% Low) are weak in the selected AOSE

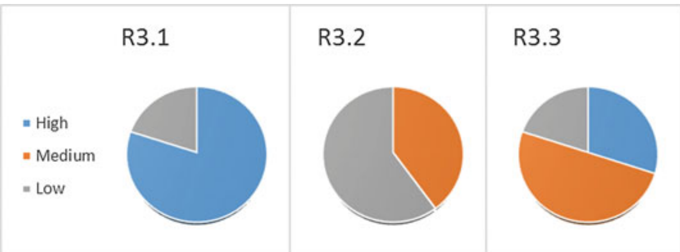


Fig. 8 Proportions of items from requirement 3 (CR3) of AOSE methodologies for CPPS

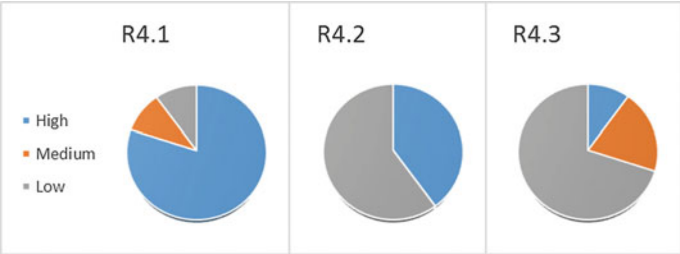


Fig. 9 Proportions of items from requirement 4 (CR4) of AOSE methodologies for CPPS

methodologies. It is possible to improve the demanded items through understanding the complexity of human behavior and its corresponding integration into manufacturing systems in a predictable way.

4.5 Further Discussion of Requirements Proportions of AOSE for CPPS

Giving all data from Table 3, Fig. 10 shows the average values of main AOSE's reliability to apply CPPS and its global proportions.

First, Part A indicates the percentage values of coverage from every item of all requirements (R1.1–R4.3). Second, Part B shows AOSE under every requirement (CR1–CR4) and the coverage percentage. Finally, Part C indicates the general average value of AOSE coverage for all requirements.

From Fig. 10, an important issue of selected AOSE methodologies for CPPS is that the mainstream group (CR1–CR3) indicates at least 50% coverage of the requirements (58% average value in Fig. 10, Part A and Part B). The reason is that CR2 requirement's items are the ones with the maximum degree of coverage (around 66%). In the same way, CR4 requirement's items are the ones with the minimum level of coverage (it is 48% in Fig. 10, Part C). Therefore, to grow the overall average value, it is essential that furthermost moderate requirements items (R1.2, R1.3, R2.7, R3.2, R4.2, and R4.3) take delivery of specific attention since main AOSE methodologies selected do not contemplate them considerably yet.

5 Conclusion and Research Roadmap

The analyses done in this paper evaluate the approaches for implementing CPS in the manufacturing process. As discussed in the context, a CPPS could be considered as MAS and it could develop through any AOSE methodologies. As result, CPPS could have better flexibility, adaptability, and creativity due to agent-based

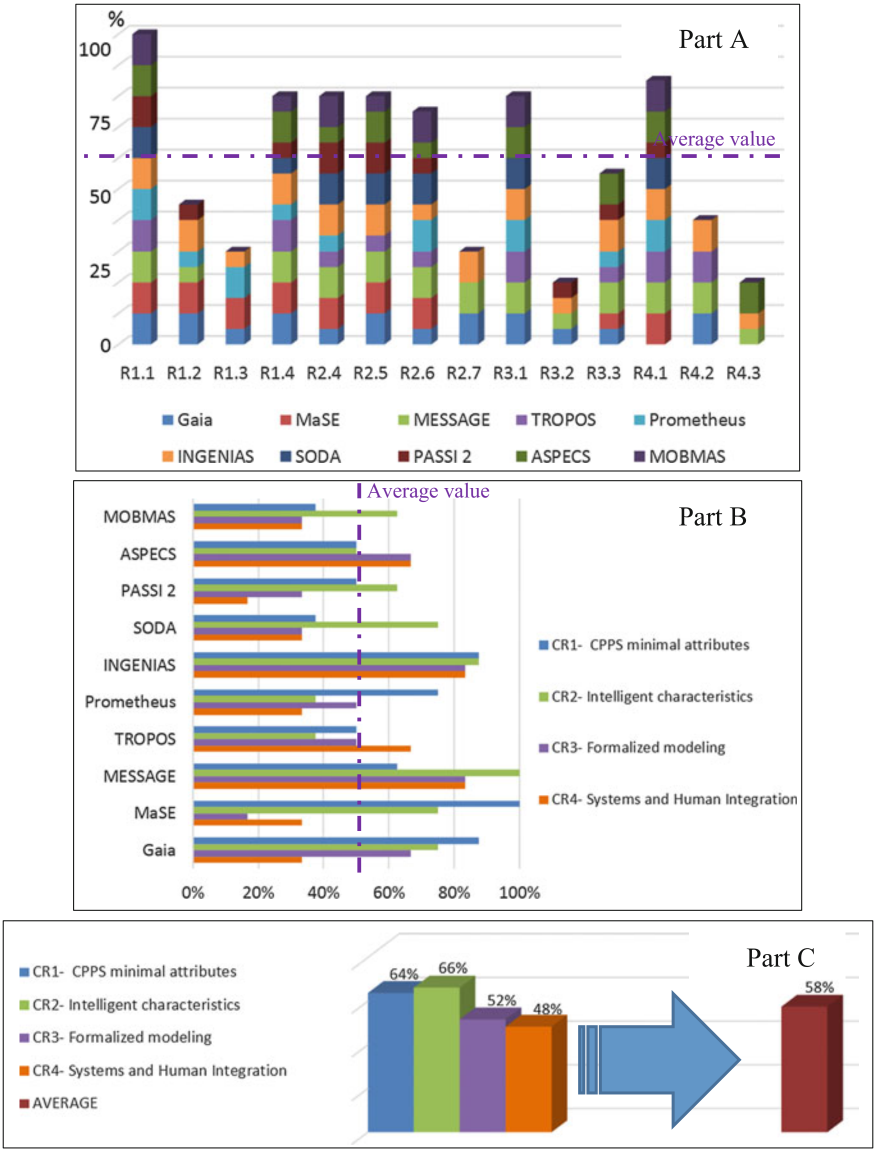


Fig. 10 Average value of main AOSE's reliability to apply CPPS (Part A: specific items coverage; Part B: specific requirements coverage; Part C: average values of AOSE coverage)

negotiation. However, first, it is crucial to consider four big requirements (CR1-CR4) that it will be necessary to identify for apply CPPS with MAS.

In the MAS approach, the essential issue of applying CPPS is how to implement AOSE methodologies reasonably. Compared with [2], this paper comes up with

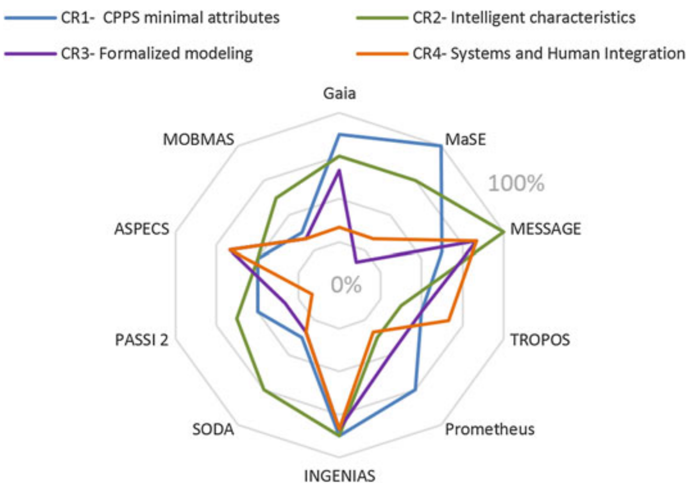


Fig. 11 General descriptive requirements' coverage from main AOSE for CPPS

quantitative proportional reliabilities focus on manufacturing systems. According to the previous discussion, AOSE could be a better option to support CPPS requirements (CR1–CR4) with 58% coverage.

Figure 11 shows generic descriptive information on requirements' coverage from main AOSE for CPPS.

Figure 11 identifies disadvantages and advantages of AOSE methodologies to apply CPPS and determine probable reasons. The majority of AOSE disadvantages for CPPS should be tackled in future to improve the performance. The results of each method must be evaluated through criteria metrics according to the imposed requirements. Thus, it is beneficial for future CPPS developers to define desired benefits and compare with potential providers. For example, either modularity [28] or flexibility [29] is one of the top goals of CPPS, and both of them require metrics to estimate reliable results.

References

1. Borangiu, T.: Studies in Computational Intelligence 544 Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics. https://doi.org/10.1007/978-3-319-04735-5 (2015)
2. Morariu, O., Morariu, C., Borangiu, Th: Shop-floor resource virtualization layer with private cloud support. J. Intell. Manuf. **27**(2), 447–462 (2014)
3. Christensen, J.H.: Holonic Manufacturing System: Initial Architecture and Standards Directions, pp. 1–20 (1994)
4. Trentesaux, D.: Distributed control of production systems. Eng. Appl. Artif. Intell. **22**, 971–978 (2009)

5. Cruz Salazar, L.A.: Estructura de control para procesos de producción desde el paradigma de los sistemas holónicos de manufactura. Master thesis, Faculty of Electronics and Telecommunications, University of Cauca. https://doi.org/10.13140/rg.2.2.29031.04007 (2017)
6. Mattik, I., Amorim, P., Günther, H.-O.: Hierarchical scheduling of continuous casters and hot strip mills in the steel industry: a block planning application. Int. J. Prod. Res. **7543**, 1–16 (2013)
7. Chung, S.H., Chan, F.T.S., Chan, H.K.: A modified genetic algorithm approach for scheduling of perfect maintenance in distributed production scheduling. Eng. Appl. Artif. Intell. **22**, 1005–1014 (2009)
8. Zhang, M.T., et al.: Hierarchical capacity planning with reconfigurable kits in global semiconductor assembly and test manufacturing. IEEE Trans. Autom. Sci. Eng. **4**, 543–552 (2007)
9. Wannagat, A., Vogel-Heuser, B.: Agent oriented software-development for networked embedded systems with real time and dependability requirements in the domain of automation. In: Proceedings of 17th World Congress IFAC, pp. 4144–4149 (2008)
10. Saumeth, C.K.J., Pinilla, T.F., Fernández A.A., Muñoz, A.D.J., Cruz Salazar, L.A.: Sistema Ciber-Físico de una CNC para la producción de circuitos impresos. In 4th Congreso Internacional de Ingeniería Mecatrónica y Automatización - CIIMA'15, pp. 154–155, Fondo Editorial EIA (2015)
11. Rüßmann, M., et al.: Industry 4.0. The Future of Productivity and Growth in Manufacturing. Boston Consulting, pp. 1–5 (2015)
12. Catalán, C., Serna, F., Blesa, A.: Industria 4.0 en el Grado de Ingeniería Electrónica y Automática, Actas las XXI Jornadas la Enseñanza Univ. Informática, pp. 327–332 (2015)
13. Wang, L., Haghighi, A.: Combined strength of holons, agents and function blocks in cyber-physical systems. J. Manuf. Syst. **40**, 25–34 (2016)
14. Beydoun, G., Low, G., Tran, N., Bogg, P.: Development of a peer-to-peer information sharing system using ontologies. Expert Syst. Appl. **38**, 9352–9364 (2011)
15. Briickner, S., Wyns, J., Peeterst, P., Kollingbaum, M.: Designing Agents for Manufacturing Control (1998)
16. Bettenhausen, K., Kowalewski, S.: Cyber-Physical Systems : Chancen und Nutzen aus Sicht der Automation, VDI/VDE-Gesellschaft Mess- und Autom., pp. 1–12 (2013)
17. Zhang, Y., Mao, S., Yang, L.T., Chen, T.M.: National Science Foundation: Cyber-Physical Systems (CPS) Program Solicitation NSF 17-529 (2016)
18. Lee, E.A.: The past, present and future of cyber-physical systems: a focus on models. Sensors (Switzerland) **15**, 4837–4869 (2015)
19. Kornerup, J.: A vision for overcoming the challenges of building cyber-physical systems. In: National Science Foundation Workshop on Critical Physical Infrastructures, pp. 1–3 (2006)
20. McFarlane, D., Giannikas, V., Wong, A.C.Y., Harrison, M.: Product intelligence in industrial control: theory and practice. Annu. Rev. Control **37**, 69–88 (2013)
21. Lu, Y., Morris, K., Frechette, S.: Current Standards Landscape for Smart Manufacturing Systems. https://doi.org/10.6028/nist.ir.8107 (2016)
22. Vogel-Heuser, B., Diedrich, C., Pantförder, D., Göhner, P.: Coupling heterogeneous production systems by a multi-agent based cyber-physical production system. In: Proceedings of 12th IEEE International Conference on Industrial Informatics, INDIN'14, pp. 713–719 (2014)
23. Leitão, P., et al.: Smart agents in industrial cyber physical systems. Proc. IEEE **104**, 1086–1101 (2016)
24. Ribeiro, L., Barata, J., Onori, M., Hoos, J.: Industrial agents for the fast deployment of evolvable assembly systems. Ind. Agents Emerg. Appl. Softw. Agents Ind. **301** (2015)
25. Dam, H.K., Winikoff, M.: Towards a next-generation AOSE methodology. Sci. Comput. Program. **78**, 684–694 (2013)
26. Case, D.M.: Engineering Complex Systems with Multigroup Agents (2015)

27. Vogel-Heuser, B., Fischer, J., Feldmann, S., Ulewicz, S., Rösch, S.: Modularity and architecture of PLC-based software for automated production systems: an analysis in industrial companies, p. 131. J. Syst, Softw (2017)
28. Dias-Ferreira, J., Ribeiro, L., Akillioglu, H., Neves, P., Onori, M.: BIOSOARM: a bio-inspired self-organising architecture for manufacturing cyber-physical shopfloors. J. Intell. Manuf. 1–24 (2016)
29. Trentesaux, D., Borangiu, T., Thomas, A.: Emerging ICT concepts for smart, safe and sustainable industrial systems. J. Comput. Ind., Editorial of Special Issue "Emerging ICT concepts for smart, safe and sustainable industrial systems", **81**, pp. 1–10 (2016)

Empowering a Cyber-Physical System for a Modular Conveyor System with Self-organization

José Barbosa, Paulo Leitão and Joy Teixeira

Abstract The Industry 4.0 advent, advocating the digitalization and transformation of current production systems towards the factories of future, is introducing significant social and technological challenges. Cyber-physical systems (CPS) can be used to realize these Industry 4.0 compliant systems, integrating several emergent technologies, such as Internet of Things, big data, cloud computing and multi-agent systems. The paper analyses the advantages of using biological inspiration to empower CPS, and particularly those developed using distributed and intelligent paradigms such as multi-agent systems technology. For this purpose, the self-organization capability, as one of the main drivers in this industrial revolution is analysed, and the way to translate it to solve complex industrial engineering problems is discussed. Its applicability is illustrated by building a self-organized cyber-physical conveyor system composed by different individual modular and intelligent transfer modules.

Keywords Cyber-physical systems · Multi-agent systems · Self-organization

1 Introduction

Manufacturing remains a key force to drive the world's economic growth. In particular, the manufacturing sector employed 31 million persons in EU-27's in 2009, and has generated EUR 5 812 billion of turnover and EUR 1 400 billion of value added [6]. In the last years, industrial manufacturing companies are facing strong

J. Barbosa (✉) · P. Leitão · J. Teixeira
Polytechnic Institute of Bragança, Campus Sta Apolónia, 5300-253 Bragança, Portugal
e-mail: jbarbosa@ipb.pt

P. Leitão
LIACC–Artificial Intelligence and Computer Science Laboratory, Rua Campo Alegre 1021, 4169-007 Porto, Portugal
e-mail: pleitao@ipb.pt

J. Barbosa
INESC-TEC, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_12

pressures by customers that demand more customized and high-quality products [7], which requires the implementation of more flexible, reconfigurable and responsive production systems to maintain their competitiveness levels.

The adoption of Industrie 4.0 principles [11], known as the 4th industrial revolution and characterized by the digitization of traditional factories, allows companies to be more competitive facing the continuous pressure imposed by global markets and demanding customers. An important remark is that this revolution will bring significant benefits with only about 40–50% of equipment replacement [3]. The use of Cyber-Physical Systems (CPS) [1] can be seen as the backbone platform to implement the Industry 4.0 vision complemented with disruptive technologies, which according to the the McKinsey's report [3] can be grouped into four clusters: (i) data, computational power and connectivity (e.g., big data, Internet of Things (IoT), Machine-to-Machine (M2M) and cloud technologies), (ii) analytics and intelligence (e.g., data mining, artificial intelligence and machine learning), (iii) human machine interaction (e.g., virtual and augmented reality), and (iv) digital to physical conversion (e.g., additive manufacturing and advanced collaborative robotics).

Multi-agent systems (MAS) [8, 14] is a proper technology to implement distributed intelligence in CPS solutions, complemented with other emergent technologies such as Service-oriented Architectures, cloud computing and big data analytics. MAS allows the development of large-scale complex engineering problems by decentralizing the control functions over distributed and intelligent software agents, that cooperate together to achieve the system goals. This kind of systems needs to be modular, flexible, robust and scalable, but also be aware of reconfiguration, adaptation and evolution in a very fast, automatic mode.

This challenge can be addressed by enhancing these systems with simple but powerful biological mechanisms, which are working in nature since millions of years, to be more responsive and agile to emergence. Self-organization is one powerful concept that can be found in several domains, such as biology (e.g., the ants foraging and birds flocking), chemistry (e.g., the Belousov-Zhabotinsky reaction), physics (e.g., the 2nd thermodynamics law) and social organization (e.g., traffic and pedestrian walk in crowed environments). Basically, self-organization is a process of evolution where the development of emergent and novel structures takes place primarily through the system itself, and normally triggered by internal forces. The challenge is to understand the principles of this concept and to translate them, case-by-case, to solve the engineering problems (note that in some situations, the application of these methods didn't reach the desired results since the concepts were simply copied).

Having this in mind, this paper discusses the benefits of enhancing agent-based systems with self-organization capabilities. In particular, the deployment of a self-organized CPS for a modular conveyor system will be described to show that the plugability and the dynamic system reconfiguration can be performed automatically and on-the-fly, i.e. without the need to stop, re-program and re-start the system. This issue is completely aligned with the requirements associated to the Industry 4.0 initiative.

The rest of the paper is organized as follows: Sect. 2 overviews the upcoming challenges of CPS and introduces self-organization as an aggregation factor for the CPS implementation. Section 3 describes the case study based on modular Fischertechniks conveyors and Sect. 4 describes the engineering of the self-organized CPS. Section 5 presents the self-organization mechanisms aiming at the plugability and reconfiguration of conveyors on the fly. Lastly, Sect. 6 rounds up the paper with the conclusions.

2 Self-organization in Cyber-Physical Systems

2.1 Overview of the Cyber-Physical System Concept

CPS is a paradigm, initially introduced in 2006 by a working group composed of experts from the USA and European Union, which refers a network of interacting computational and physical devices [12], suitable to build complex and large-scale systems. This paradigm advocates the co-existence of cyber and physical elements [15] with a common goal to build complex systems following the principle of "system of systems", which simplifies their understanding, design and deployment. CPS are being applied to different domains, namely smart industrial production, smart electrical grids, smart logistics, smart traffic control, smart e-health, and smart cities and buildings.

As stated by [15], the adoption of CPS in industrial environments is faced by several challenges, namely:

- *CPS Capabilities*, comprising amongst others the modularization and servitisation of CPS systems and the consideration of advanced (big) data analytics.
- *CPS Management*, including security and trust in the management of large scale CPS.
- *CPS Engineering*, comprising methods and tools for the CPS life-cycle support, including the design, development and deployment.
- *CPS Ecosystems*, focusing on the design and deployment of collaborative, autonomic, self-organised and emergent CPS, as well as the integration of Humans in the Loop.
- *CPS Infrastructures*, related to interoperability services, mitigation and migration strategies to support CPS infrastructures.
- *CPS Information Systems*, considering artificial intelligence and the transformation of data and information analytics to actionable knowledge.

Among the identified challenges, the use of bio-inspired mechanisms, and particularly self-organization to support the dynamic evolution and adaptation to condition changes, is a promising issue to empower the deployment of such CPS solutions.

2.2 Self-organization as the CPS Aggregation Factor

Evolution is the process of change, namely the development, formation or growth over generations, leading to a new, more advanced or complex form [13]. Self-organization is one approach to achieve the dynamic system evolution, being defined as a process of evolution where the development of new, more advanced and complex structures takes place primarily through the system itself, and normally triggered by internal driving forces. Different types of self-organization can be observed in nature, e.g., stigmergy (e.g., used by ants), decrease of entropy (e.g., represented by the 2nd law of thermodynamics) and autopoiesis (e.g., found in the cells reproduction), each one defining different driving forces to trigger the self-organization process.

Research in self-organization mechanisms is taking much attention, being applied to different domains, such as economics, sociology, computing, robotics and manufacturing. Particularly in the manufacturing domain, self-organization allows the dynamic and on-the-fly evolution and re-configuration of the organizational control structure, supporting the agile reaction to unpredictable condition changes. For this purpose, the self-organization concept can be translated for the development of self-organized CPS, which allows achieving truly reconfigurable systems that address the current industrial requirements. These self-organized CPS allow reaching the requirements of flexibility, robustness, adaptation, reconfigurability and agility, in a simple and intrinsic manner, with the system behaviour emerging from the interaction among individuals.

These systems are traditionally more difficult to design since complexity also comes from the non-linear interactions among individuals involving amplification and cooperation, and the sensitivity to initial conditions (i.e. the butterfly effect). In fact, the resulting behaviour is difficult to predict since the large number of non-linear interactions may cause a large number of possible non-deterministic ways in which the system can behave. For this purpose, the self-organization process should be properly controlled to guarantee that expected properties will actually emerge, and not expected and not desired properties will not emerge. Additionally, in such dynamic and self-organized systems, the existence of regulation mechanisms are crucial to control the system's nervousness and to maintain the system in a stable state, avoiding the increase of entropy and chaos.

Self-organization was used in several works in manufacturing the domain, namely in P2000+ [5], ADACOR [16], PROSA + ants [19], AirLiquide [17] and ADACOR^2 [2]. This last approach combines a behavioural self-organization perspective to ensure the smooth system's evolution (aligned with the Darwin's theory of evolution of the species) and the structural self-organization perspective to support the drastic evolution episodes (aligned with the punctuated equilibrium theory). However, the number of practical applications running in industrial environments is reduced or uses weak self-organization implementations. A significant work should be performed to disseminate the potentialities of using self-organization in large and complex industrial CPS.

3 Modular Cyber-Physical Conveyor System

The case study considered in this work is related to a conveyor system composed by a set of modular Fischetechnik conveyors, each one having the same structure and offering the same functionalities, i.e. to convey parts from the input to the output position. Physically, each individual conveyor is composed by a belt operated by a 24 V DC motor. The detection of parts at the beginning and ending positions is achieved by means of independent photo-electric barrier sensors. Similarly to the motor, the sensors operate at the nominal voltage of 24 V, making it industrially compatible, e.g., allowing their direct connection to a Programmable Logic Controller (PLC). In such system, the conveyor starts its motor when the part arrives to the output sensor of the previous conveyor, and stops its motor when the part arrives to the input sensor of the next conveyor.

The logical control of the aforementioned modular conveyor system can be implemented using several approaches. A traditional choice would be to use a centralized IEC61331-3 [9] control program running in a PLC to regulate the behaviour of the overall conveyor system, to provide scalability and reconfiguration. In particular, the easy re-configuration of the conveyor system, e.g., add a new conveyor or switching the order of conveyors modules, is complex and time-consuming. The use of the IEC61499 standard [10] presents interesting features, such as its distributed nature, but lacks the support for the intelligent and autonomous decision, which is crucial for the development of self-organization mechanisms. The challenge is to use MAS technology to achieve the self-organization and reconfiguration on-the-fly, i.e., without the need to stop, re-program and re-start the system components, which can not be easily achieved by traditional approaches.

In this work, the logical control of the conveyor module uses agent technology deployed into the Raspberry Pi boards. The Raspberry Pi control board is powered by a standard USB cable, and therefore uses a 5 V power supply while its General Purpose Input Output (GPIO) ports are 3.3 V compatible. This requires the need to develop an interface board, also commonly named as “shield”, mainly responsible for:

- Supply the power to the conveyor belt and provide an isolated supply for the Raspberry Pi board.
- Connect physically the Raspberry Pi (processing part) and the conveyor belt (physical part).
- Indicate, by using LEDs, the system operation and the GPIO usage.

In this way, the individual conveyor system is divided into two symbiotic parts, namely the logical part and the physical part. The logical part is provided by the processing capabilities offered by the agents running in the Raspberry Pi, constituting the *cyber* level, while the *physical* part is provided by the functions offered by the group formed by the conveyor, motor and sensors. By this, and as depicted in Fig. 1, the bundle constitutes a cyber-physical component, which, by combining different similar ones will constitute a cyber-physical system for the conveyor system.

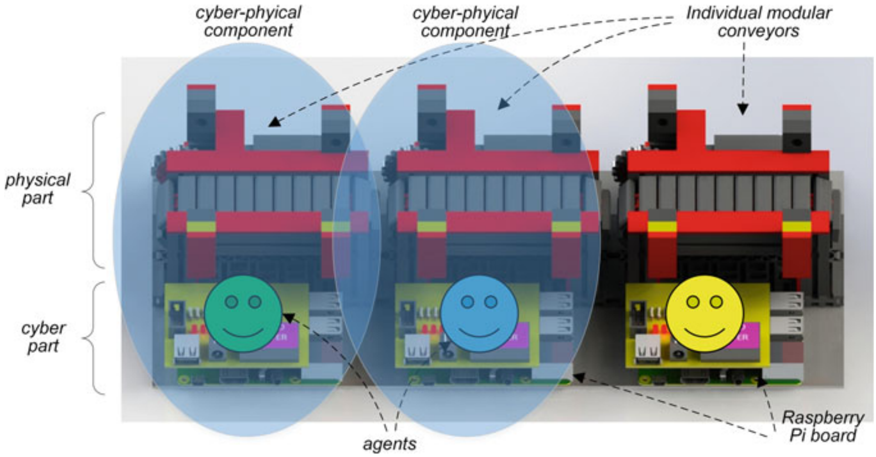


Fig. 1 Individual conveyor as a cyber-physical component

4 Engineering the Modular and Self-organized Conveyor System

This section presents the engineering of the modular conveyor system composed of several similar individual conveyors, and describes the technical issues regarding the deployment of agents into the Raspberry Pi control boards.

4.1 Designing the Agents

The JADE (Java Agent DEvelopment Framework) [4] platform was used to implement the agents that will control the individual conveyors. One of the most important features of implementing the logical control by using the agent technology is the possibility of programming-once and deploying multiple times. Nevertheless, this interesting feature is only fully exploited when the agents are properly designed to be as generic as possible. In this way, the behaviour of each agent controlling an individual conveyor comprises a simple initiation process that involves the registration of the agent's skills in the yellow pages, the search of agents with similar skills and a general announcement of the conveyor presence in the system. This phase is finalized by launching the behaviours that are responsible to handle the message exchange, the internal logics and the self-organization process.

In the logical perspective, the behaviour of each agent relies on the control of the motor of its conveyor according to the input and output sensors and the synchronization with the behaviour of precedent and posterior conveyors, as represented by the Petri nets [18] model illustrated in Fig. 2. In fact, each conveyor must inform

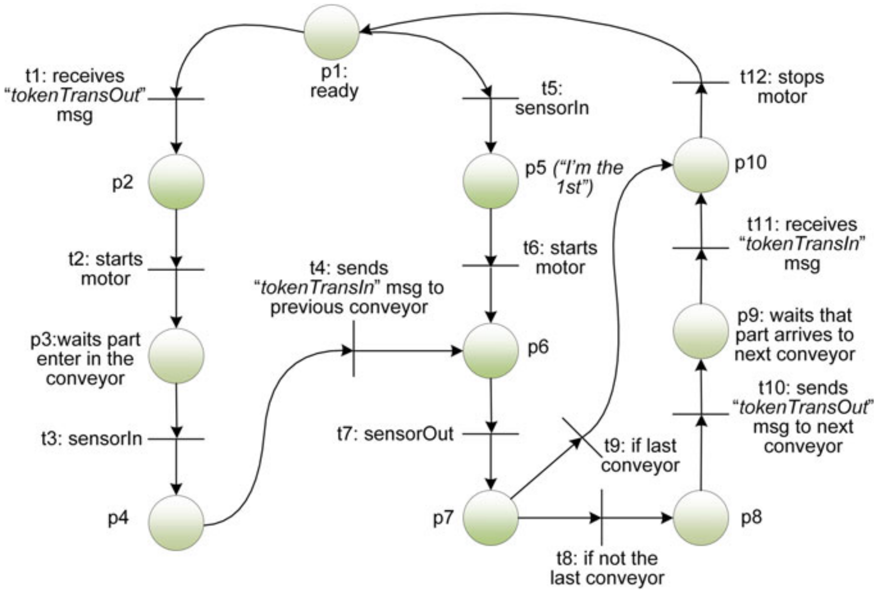


Fig. 2 Logic control for an individual conveyor agent

the adjacent conveyors in a two-fold manner. When the conveyed part is at the input sensor, the conveyor must inform the previous one (if any) that it has the part possession, allowing the previous conveyor to stop its motor. Similarly, when the part is at the output sensor, the conveyor must inform the successive conveyor (if any) that it should start its motor in order to properly receive the part.

The management of this collaborative process among the conveyors is performed by the agents through the exchange of messages following the FIPA Agent Communication Language (ACL). The *tokenTransIn* message is used to inform the agent that the part is at the input sensor of the posterior conveyor and the *tokenTransOut* message is used to inform the agent that the part is at the output sensor of the previous conveyor. These messages contain the reference to the *token*, i.e. the indication of the conveyor current position, which the receiving agents use to check if the token passage concerns them, and if yes, stop or start the motors accordingly.

The agents exchange other types of messages during their cooperation processes, namely the *informIAmAlive*, *thereIsASwap*, *swapTokenFound* and *IAmLeaving*. These messages, governing several phases and situations in the conveyor behaviour, will be explained in details during the description of the self-organization mechanisms.

4.2 Cold Start of the System

At system set-up, none of the agents is aware of its position in the system sequence, which requires the execution of a distributed procedure to determine the sequence of the individual conveyors. When a part is placed at a given conveyor, and the correspondent agent doesn't know its current position, the agent assumes the *token* value 1 (first conveyor in the system). Then, when the part reaches the output sensor, a message is broadcasted to all conveyors in the system. At this stage, the conveyors that have the posterior token or don't know their positions will start their motors. Note that the conveyors that already know their positions and are not the subsequent conveyors will ignore the message. The agent that receives the part and doesn't know its current position in the sequence will update it accordingly to the passed token value. This procedure is repeated until the part reaches the last conveyor.

4.3 Deployment of Agents in Raspberry Pi

After the implementation phase of the agents, these must be deployed and executed in the Raspberry Pi boards. Due to the JADE inherent features, one main container must be initiated before the agents' execution to contain the agent platform. Despite the fact that one of the Raspberry Pi boards could be selected to host this main container, a cloud-based approach was chosen to host the JADE agent-based platform. This means that no central governing mechanism is deployed into the cloud, i.e. the system is governed using a decentralized self-organization mechanism without providing any system topology, size or conveyor position to the agents. This decision allows reaching a more flexible approach permitting to remove any conveyor, releasing the user from the constant need of ensuring the main-container up-time.

The deployment of agents in the Raspberry Pi is straightforward, being only necessary to upload the agent package (i.e. the .jar file containing the agent instantiation) into a system directory. Despite this, several preparatory work in the Raspberry Pi board is advised to be performed, namely:

- The installation of a Java Run-time Environment (JRE).
- The installation of a Java API allowing the agent to access the GPIOs and/or to system information.
- The definition of the environment variables (to simplify the agent's execution from any point in the system).

The implementation uses the PI4J Java API (http://pi4j.com/) that bridges the Raspberry Pi kernel into Java compliant methods. Besides to allow the control of GPIOs, this API also allows the access to the serial communication to gather the system/network information and to create of event listeners. An important remark: to have access to the hardware in the Raspberry Pi board, the agent needs to have *root* rights.

The agent's access to the hardware is configured during its initialization behaviour where the required GPIOs are provisioned and configured properly, using the following excerpt of code:

```
myMotor = gpio.provisionDigitalOutputPin
            (RaspiPin.GPIO_04, PinState.LOW);
outSensor = gpio.provisionDigitalInputPin
            (RaspiPin.GPIO_05);
inSensor = gpio.provisionDigitalInputPin
            (RaspiPin.GPIO_06);
```

Listeners are also added to govern the agent's actions triggered by the change of state in the input and output light-sensors. The following excerpt of code exemplifies the listener that was implemented for the *inSensor*.

```
inSensor.addListener(new GpioPinListenerDigital() {
  @Override
  public void handleGpioPinDigitalStateChangeEvent(
             GpioPinDigitalStateChangeEvent event) {
    // code to be executed by Raspberry Pi board when
    // the sensorIn signal changes its state goes here
  }
});
```

5 Mechanisms for Plugability and Self-organization

An important issue during the operation of the modular cyber-physical conveyor is to ensure the presence of mechanisms that support the plugability and self-organization of the conveyor system on-the-fly, i.e. without the need to stop, re-program and restart the individual components. For this purpose, this section details the implementation of self-organization mechanisms to support the system's operation in evolvable environments, namely adding a new individual conveyor, removing a broken individual conveyor or swapping two individual conveyors. The overall self-organization mechanism is then built by the composition of few simple rules (similarly to what happens in nature).

5.1 Plug-In and Plug-Out of Individual Conveyors

The plug-in of a new conveyor can happen at different locations along the sequence, being required to design a simple mechanism for the automatic and decentralized detection of the conveyor position, as illustrated in Fig. 3.

Initially, the new conveyor agent registers its skills in the DF service and sends an *informIAmAlive* message to the other agents informing that it is ready to work.

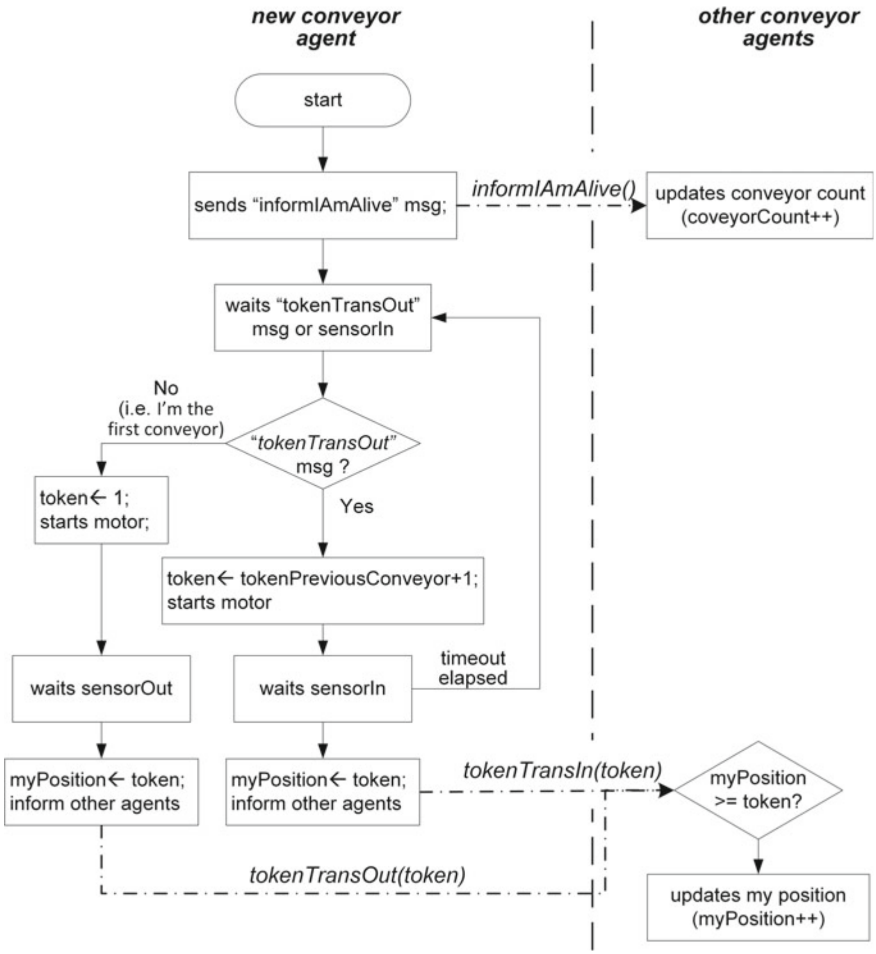


Fig. 3 Mechanism for the automatic plug-in of a new conveyor

The other conveyor agents receiving this message are able to update the number of conveyors placed in the system. After this set-up phase, the new conveyor agent is waiting for the occurrence of one of two situations: (i) the arrival of a part at the sensorIn sensor or, (ii) the arrival of a *tokenTransOut* message. The occurrence of the first case, i.e. the detection of a true signal at the input sensor of the conveyor without receiving any *tokenTransOut* message, means that the new conveyor is placed in the beginning of the conveyor sequence, and after starting its motor, the conveyor agent will send a *tokenTransOut* message when the part reaches its output sensor. In the second case, i.e. the arrival of the *tokenTransOut* message, the conveyor agent starts its motor and waits that the part arrives to its input sensor. If it occurs before a time-out, it means that the new conveyor is placed after the conveyor that has sent the

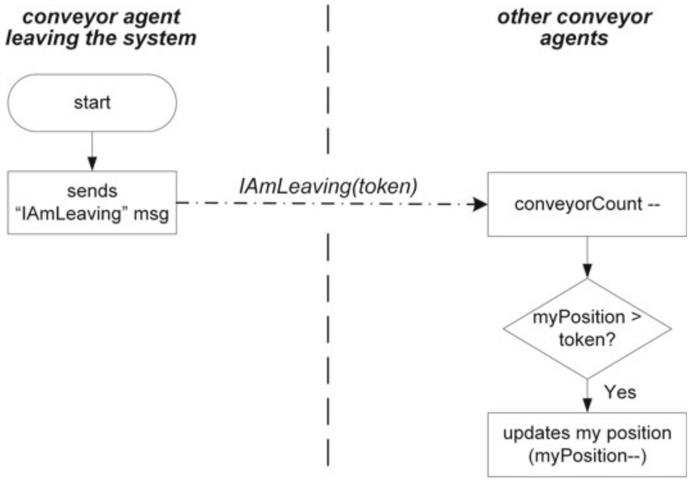


Fig. 4 Mechanism for the automatic plug-out of conveyors

tokenTransOut message. In this case, the new conveyor agent updates its location and sends a *tokenTransIn* message to the previous conveyor agent and a *tokenTransOut* message to the other conveyor agents when the part reaches its output sensor. In this case, only the conveyor agents that are located after the introduced new conveyor will update their positions.

If the new conveyor agent detects that it is located at the end of the sequence, it will stop its motor when the part reaches the output sensor and not when the *tokenTransIn* message from the posterior conveyor agent has arrived. In this case, the conveyor agents only update the individual conveyors counting without affecting the other agents.

Similarly, the removal of a conveyor is broadcasted by the leaving conveyor agent before its removal, as illustrated in Fig. 4.

The conveyor agents will decrease by one the number of conveyors placed in the system (i.e. decreasing the variable *conveyorCount*) and those placed after the removed conveyor will downward their locations by one position.

5.2 Change the Order Sequence of Conveyors

The swap mechanism requires a slightly different approach since the conveyor counting is kept and the new positions must be discovered. The self-organization mechanism to support the conveyor swap comprises several stages, as depicted in Fig. 5.

The first step is related to the detection phase. In normal operation, when a part is at the output sensor, a message is broadcasted to all the agents and is processed by the succeeding conveyor agent and discarded by the rest. When the succeeding

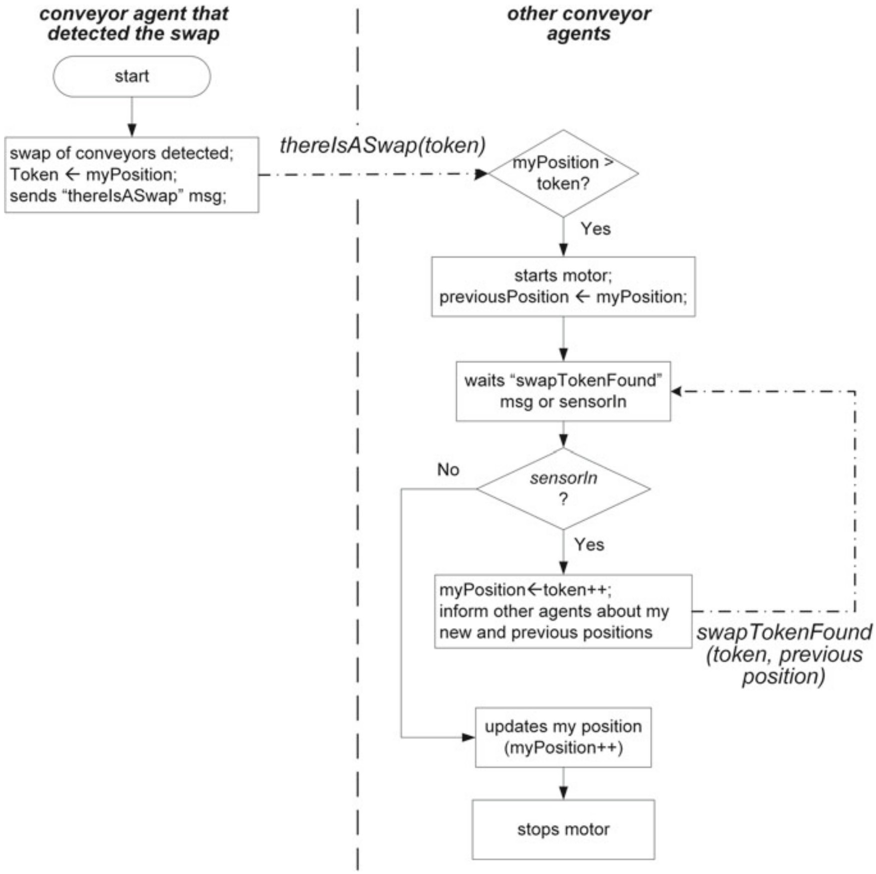


Fig. 5 Mechanism for the automatic swap of conveyors

conveyor agent detects that the piece has not reached by its input sensor (by means of a timeout), it broadcasts the *thereIsASwap* message that warns for a possible order change (containing the indication of the current position).

All the conveyor agents that have a token higher than the received position will start their motors (naturally guaranteeing that they are in a valid situation, e.g., without having a part to convey). Afterwards, when a conveyor agent receives the part at its input sensor, it will update its current system position and broadcast a *swapTokenFound* message with its previous and new positions (updated with the position received during the *thereIsASwap* message). The other conveyor agents that receive this message only update their new positions.

This process is repeated every time a possible conveyor order has changed, governing, in a decentralized and self-organized manner, the system behaviour. The use of agents and self-organization principles allow, in a distributed and decentralized manner, to govern this conveyor system using a simple set of rules. The absence of a central control logic node also increases the system robustness and scalability by eliminating single-point of failure situations. Additionally, the use of agents allow reaching truly the "plug-and-produce" concept.

6 Conclusions

This paper describes a self-organized and modular CPS composed by several several individual cyber-physical conveyors. Each individual cyber-physical conveyor is composed by a conveyor belt, constituting the physical part, and a logical part, implemented using agent technology and deployed in a Raspberry Pi board. Arranging together different individual cyber-physical conveyors allows conveying parts from an initial position to the final position by means of cooperation between all the conveyors, in a "system of systems" perspective.

A simple self-organization mechanism was deployed in a distributed manner and with MAS technology to govern the system operation. This mechanism, comprising several simple rules, allowed successful system operation, including the addition and/or removal of conveyors, or even the change of the sequence of conveyors on-the-fly, maintaining the system operability.

This CPS platform was used in the practical learning classes of the summer schools on intelligent agents in automation, held in Lisbon, Portugal in 2015 and afterwards in Linköping, Sweden in 2016, providing the hands-on experience to the participants on deploying agents and developing simple self-organization mechanisms.

The described self-organized CPS is a simple example that shows the potentialities of applying CPS and self-organization concepts to industrial environments. The future work considers the further refinement of the described self-organized mechanisms, namely refining the configuration process where diverts and convergences are needed.

References

1. ACATECH: Cyber-physical systems: driving force for innovation in mobility, health, energy and production. Technical report, ACATECH – German National Academy of Science and Engineering, Dec 2011
2. Barbosa, J., Leitão, P., Adam, E., Trentesaux, D.: Dynamic self-organization in holonic multi-agent manufacturing systems: the ADACOR evolution. Comput. Ind. **66**, 99–111 (2002)
3. Bauer, H., Baur, C., Camplone, G., et. al.: Industry 4.0: how to navigate digitization of the manufacturing sector. Technical report, McKinsey Digital (2015)
4. Bellifemine, F., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. Wiley (2007)
5. Bussmann, S., Schild, K.: Self-organizing manufacturing control: an industrial application of agent technology. In: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS'00), pp. 87–94 (2000)

6. EFFRA: Factories of the future, multi-annual roadmap for the contractual PPP under H2020. Technical report, European Commission (2013)
7. ElMaraghy, H.: Flexible and reconfigurable manufacturing systems paradigms. Int. J. Flex. Manuf. Syst. **17**, 261–271 (2006)
8. Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley (1999)
9. IEC—International Electrotechnical Commission: IEC 61131: Programmable controllers—Part 3: Programming Languages (2012)
10. IEC—International Electrotechnical Commission: IEC 61499: Function Blocks—Part 1–4 (2012)
11. Kagermann, H., Wahlster, W., Helbig, J.: Securing the future of German manufacturing industry: recommendations for implementing the strategic initiative INDUSTRIE 4.0. Technical report, ACATECH – German National Academy of Science and Engineering (2013)
12. Lee, E.A.: Cyber physical systems: design challenges. In: Proceedings of the 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'08), pp. 363–369 (2008)
13. Leitão, P.: Holonic rationale and bio-inspiration on design of complex emergent and evolvable systems. In: Hameurlain, A., Küng, J., Wagner, R. (eds.) Transactions on Large Scale Data and Knowledge Centered Systems. Lecture Notes in Computer Science, vol. 5740, pp. 243–266. Springer (2009)
14. Leitão, P.: Agent-based distributed manufacturing control: a state-of-the-art survey. Eng. Appl. Artif. Intell. **22**(7), 979–991 (2009)
15. Leitão, P., Karnouskos, S., Colombo, A.W.: Industrial automation based on cyber-physical systems technologies: prototype implementations and challenges. Comput. Ind. **81**, 11–25 (2016)
16. Leitão, P., Restivo, F.: ADACOR: a holonic architecture for agile and adaptive manufacturing control. Comput. Ind. **57**(2), 121–130 (2009)
17. Miller, P.: The Genius of Swarms. National Geographic (2007)
18. Murata, T.: Petri nets: properties. IEEE Anal. Appl. **77**(4), 541–580 (1989)
19. Valckenaers, P., Hadeli, K., Kollingbaum, M., Brussel, H., Bochmann, O.: Stigmergy in holonic manufacturing systems. J. Integr. Comput.-Aided Eng. **9**(3), 281–289 (2002)

Approaching Industrial Symbiosis Through Agent-Based Modeling and System Dynamics

Melissa Demartini, Flavio Tonelli and Filippo Bertani

Abstract Sustainable industrial systems are complex, since they exhibit both detail and dynamic complexity. Only an integrated approach is able to provide a realistic view of such complex systems providing a useful insight of their behaviour. In this paper, a hybrid approach based on Agent Based Modeling (ABM) and System Dynamics (SD) is presented in order to improve modelling insight of an industrial symbiosis (IS) context. Hybrid approaches have gained prominence overpassing limitations of traditional methodologies and tools, as well as computational advances that permit better modelling and analysis of complex systems with a particular focus on sustainability topics exploiting the strengths of both ABM and SD models, while minimizing the drawbacks. Therefore, to provide a methodological proof, an application of the proposed hybrid approach to an industrial symbiosis relevant case is presented and discussed. The methodological approach adopted in this research can be used to investigate a variety of industrial symbiosis cases providing insights usually not achievable with standard techniques and tools.

Keywords Hybrid approach · Agent based modelling · System dynamics Industrial symbiosis

1 Introduction

In the last few years, sustainability has reached great interest and has driven companies to reengineer their processes and products with the aim to achieve efficiency of materials, resources and then obtain an economic value. This change

M. Demartini (✉) · F. Tonelli · F. Bertani
Department of Mechanical Engineering, Energetics,
Management and Transportation (DIME), University of Genoa,
Via All'Opera Pia 15, 16145 Genoa, Italy
e-mail: melissa.demartini@dime.unige.it

F. Tonelli
e-mail: flavio.tonelli@unige.it

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_13

brings to a better awareness of material consumptions and production cycle, with less usage of resources as energy and water, less emissions while producing goods that are sustainable in their whole life cycle. An additional step in the sustainable development is represented by the so-called Industrial Symbiosis (IS); it concerns the collaboration between two or more industries, which, with specific agreements, supports the exchange of waste and by-products to be used as raw materials for production processes without the resort to new raw materials. Chertow defines IS as "the collaboration of individual entities to a common approach which leads to a competitive advantage that involves exchange of materials, energy, water and by-products" [1].

In this paper, a hybrid approach based on Agent Based Modeling (ABM) and System dynamics (SD) is presented in order to improve modelling insight of an industrial symbiosis context with the aim of moving beyond the static representation of environmental-economic variables and deal with the system's dynamic complexity. Specifically, IS has been selected as a proof of concept for the proposed conceptual hybrid model. Indeed, there is a need for decision making tools that will be able to simulate the IS system's response to different policies to take more informed decisions.

1.1 Agent-Based Models and Sustainability

Agent-based Models (ABM) or multi-agent systems are a class of computational models designed to simulate action and interaction between autonomous agents, which can be both individual entities and collective ones such as organizations, with the aim of studying effects at aggregate level produced by the interaction between agents on the whole system. ABM has gained prominence through new insights on the limitations of traditional assumptions and approaches, as well as computational advances that permit better modelling and analysis of complex systems and particularly in the sustainability domain. ABMs in the industrial sustainability field are emerging and various authors have identified the potential value and effectiveness, and advocated such simulation approaches for its characteristics [2]. The key advantage is the ability to take into account heterogeneity and behavioural interactions, which can lead to emergent behaviour that would not be obvious or might be very difficult to foresee in an aggregate model as it could occur in the current manufacturing networks [2].

1.2 System Dynamics and Sustainability

System dynamics (SD) is a methodology developed at the end of 1950s at M.I.T. of Boston, and afterward it spread in the University context. Often human beings operate in systems characterized by high level of dynamic complexity; these systems could be connected to sustainability, physic, ecology, sociology and economy. SD is a representative method for measuring the long-term dynamics of complex system, which fits for measuring the dynamics of sustainability. It is a simulation method to identify behaviour changes according to the structural characteristics of a system on the basis of the causal relationships among system factors [3]. The inherent flexibility and transparency is particularly helpful for the development of simulation models for complex sustainability systems with subjective variables and parameters. Therefore, this method can consistently be used to understand sustainability discussions [4].

1.3 Hybrid Approach and Sustainability

ABM and SD are among the most important simulation available methods; both of these approaches are used to study the leverage points of complex systems. Advantages and limitations of individual methods were the motive for the emergence of integrated simulation approach. Sustainable systems are complex; they exhibit both detail and dynamic complexity, in fact they represent a form of Complex Adaptive Systems (CAS) because they involve multiple sectors and agents displaying non-linear and non-rational interacting behaviours characterized by feedbacks and time lags.

Therefore, we claim that a hybrid SD-ABM approach may potentially better address such issues in a more informative and effective way because they exploit the strengths of both models, while minimizing the drawbacks, and providing a more realistic view of such complex systems [5]. Lättilä et al., argue that using both methods they will improve the quality of the model and give more in-sights, but at the same time they highlights the need for further researches regarding the actual simulation models [6]. SD and ABM are developed around the real characteristics of the phenomenon they aim at reproducing and simulating, limiting the use of assumptions. In this way, they provide a useful platform to model non-linear phenomena, in particular, they are able to: (i) show the impact of indirect effects on the agents and components of the model; (ii) shape relations according to their governing feedback loops; (iii) internalize in agents' behaviour (ABM) or system relations (SD) the externalities linked to specific actors and situations; (iv) represent as exogenous variables and sectors those taken as exogenous by neoclassical models [7]. Thanks to these desirable characteristics, these modelling tools are able to shed greater light on the world we are living in, characterized by time lag between agents' decisions (governments, households, industries), non-rational actors, which have a specific behaviour influenced by several inputs, and which act as free riders against future generations. Moreover, providing a closer representation of reality, they are also able to show the "unintended effects" of the introduction of new policy measures, such as the rebound effect. Finally, it is possible to say that SD and ABM complement each other.

The purpose of this paper is to develop a hybrid model for IS and the main objectives are:

- Asses sustainability in the IS network;
- Simulate the IS network in order to better understand and analyse critical problems;
- Analyze how resources' consumption changes with respect to the symbiosis.

The paper is organized as follows. Section 2 introduces the Research methodology adopted for this paper. Section 3 presents the hybrid model, the practical application and the simulation scenarios. Results and discussion are given in Sect. 4. Finally, conclusions are provided in Sect. 5.

2 Methodological Approach

2.1 Introduction

A hybrid system of ABM and SD has been proposed in order to address the unique characteristics of the IS problem: (i) nonlinear properties which would not allow us using classic econometric models, (ii) positive and negative feedback which influence its behaviour, and (iii) these behaviours can be fully understood in the interactions of the models. In particular, with SD-ABM hybrid models we can model a component with SD or ABM, but run it only with the most effective one at a given time. The hybrid modeling and simulation approach are suitable to evaluate the system outputs in both macroscopic and microscopic point of view for many strategy-making [8]. IS presumes that industries collaborate intentionally and organize themselves in order to reach not only a better use of materials, but also a partnership that permits to share strategies and objectives. In this direction, the adoption of the hybrid approach for IS allows examining different strategies, creating a dynamic environment for agents, which can actively behave in the system and interact with each other [9, 10].

In the work discussed here, system dynamics have been selected for considering flows and feedback dynamics from an aggregated viewpoint [11]. SD allows revealing the trend and system-level behaviour explicitly and intuitively. On the other hand ABM has been selected because it assumes no fixed system structure and the overall system behaviour emerges from individual agent rules, making it thus a bottom-up modelling approach. The IS system has been simulated by the Anylogic tool, which provides both the agents and the SD model. The challenge is to determine:

1. Which are the individual behavioural aspects that may favour industrial symbiosis?
2. Which are the overall benefits associated with the entire industrial network?

2.2 Case Description

IS has been selected as proof of concept for the proposed conceptual hybrid model. The scope of the model is creating an eco-industrial development plan by incorporating the basic ideas of industrial symbiosis, industrial ecology and eco-industrial parks. The model considers an industrial network made up of 4 firms. Each firm produces a single main product sold on the final market. The production process requires a single input, purchased from the external supply market, and generates a single waste product, which is destined for landfill. Each firm gets revenues from selling its main product, while production costs are in the form of purchasing and waste disposal. We considered 4 industries: a manufacturer of mechanical components (MC), a steel plant (SP), a cement plant (CP) and a paper factory (PF). Table 1 shows input/output materials for each industry, while Fig. 1 depicts the available links between the 4 companies. We assume that each firm can send and receive waste from any firm. Each firm within the industrial network is modelled as an agent that decides whether or not to establish a symbiotic relationship with another firm belonging to the other industries.

Table 1 Input/output materials for each industry

Industry	Input	Output	Waste
Mechanical components	Cast iron	Pulley	Cast iron
Steel plant	Carbon/cast iron	Steel	Slag steel
Cement plant	Sand/mixed metal slag	Cement	Waste water
Paper factory	Wood pulp	Paper	Paper mill waste sludge

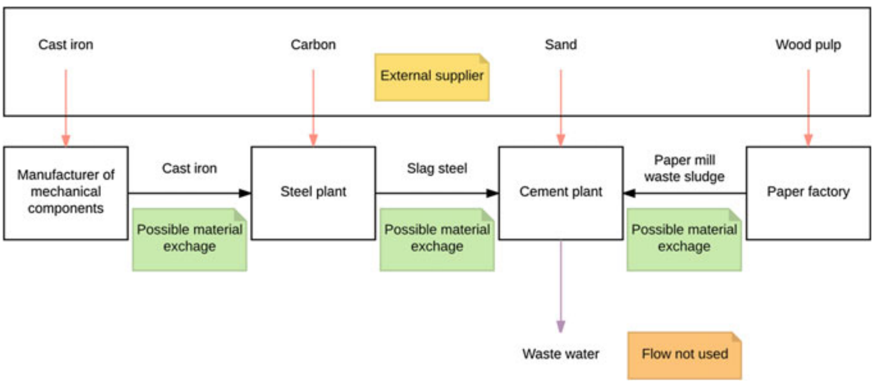


Fig. 1 Available links between the four companies

2.3 Model Development: SD + ABM

In this section the SD and ABM models are developed and described. The hybrid model was developed in AnyLogic simulation software. In general, SD modelling uses stocks and flows. Stock is the state of a system and describes its current status. The flow affects the stock and interlinks it within its system. When the dynamic model is properly developed, the quantitative representation can be simulated (see Fig. 2).

The dynamic behaviour of the system can be explained by a set of mathematical equations, next described. The dynamic behaviours of stock (such as “Raw material inventory”, “Service inventory”, “Final products inventory” and “Waste inventory”) are given by a time integral of the net inflows minus the net outflows. Due to the high number of variables, the specific mathematical formulation and process model description were illustrated for one of these models; the others reported in the Appendix.

Figure 2 shows the SD-ABM model of the manufacturer of mechanical components, who produces pulleys starting from a customer order. This process has been managed through the variable demand, which is modelled as a random walk. In fact, it consists of a succession of demands, and each demand takes in consideration the previous one. Table 2 shows the demand for each company.

After an order is received, a production order is generated and at the same time the raw material order is created. The raw material order is linked to the external supplier, which in this specific industry provides cast iron. The raw material flow provided by the external supplier is controlled by an event called “Production MC”. This event allows the material transfer only if the raw material’s level is below a specific value. This event allows to not filling up warehouses in an uncontrolled way.

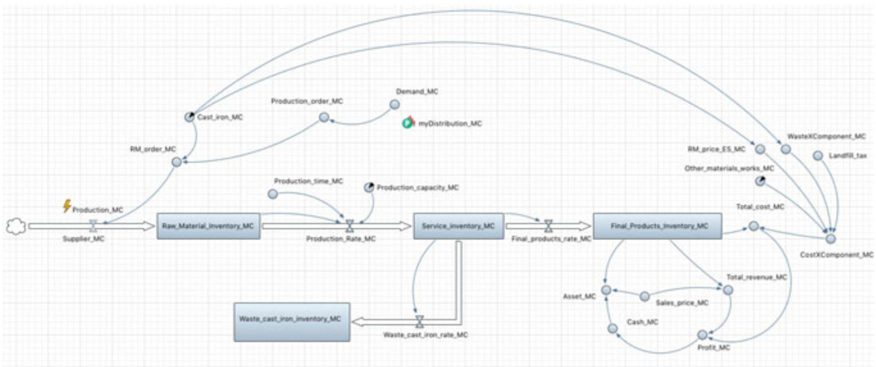


Fig. 2 Hybrid model of the manufacturer of mechanical components

Table 2 Demand of each firm

Week	1	2	3	4	5	6	7	8	9	10
Demand MC (ton)	• 3500	• 3310	• 3233	• 3180	• 2914	• 2806	• 2883	• 2915	• 2920	• 3012
Demand SP (ton)	• 28	• 18	• 24	• 16	• 25	• 18	• 26	• 23	• 16	• 26
Demand CP (ton)	• 10000	• 9680	• 9810	• 9803	• 9608	• 9892	• 10050	• 9930	• 9663	• 10108
Demand PF (ton)	• 1148	• 1000	• 1072	• 948	• 800	• 936	• 1045	• 914	• 778	• 1078

Table 3 Production times of each firm

	Week
Mechanical components	• 3
Steel plant	• 3
Cement plant	• 1
Paper factory	• 1

Table 4 Production capacities of each firm

	Ton
Mechanical components	• 4000
Steel plant	• 35
Cement plant	• 10500
Paper factory	• 1200

Table 5 Percentage of waste generated by each firm

	Percentage of waste generated (%)
Mechanical components	• 5
Steel plant	• 9
Cement plant	• 5
Paper factory	• 8

At the beginning of simulation, it is supposed that there are no symbioses between firms, and that only external suppliers provide raw materials. After the procurement phase, raw materials are stored in the raw material inventory waiting to be worked out. The production phase allows transforming raw materials into final products; this is done through constant monitoring of the production capacity, which varies for each factory. Tables 3 and 4 show production time and production capacity for each firm.

During the production process, each firm generates wastes; Table 5 shows the percentage of waste generated by each company.

Finally, for each firm an economic analysis is provided. In fact, each firm obtains revenues from selling its main product (it has been supposed all final products are sold), while production costs are the purchasing cost and waste disposal cost. Figure 3 gives details concerning revenue, cost and profit of each firm of the hybrid model. If there is no link between factories, total costs are the sum of: purchasing cost which depends by the raw materials price imposed by the external supplier, waste disposal cost which depends by the landfill tax imposed by government, and a generic cost which considers the costs of other raw materials and works with the scope of provide reasonable results. These considerations are also done for the material flow in order to respect the mass balance. Revenue is calculated by multiplying the number of final products with the sale price. Finally, there are two last variables, which are Cash and Asset; the first one shows the current level of cash in

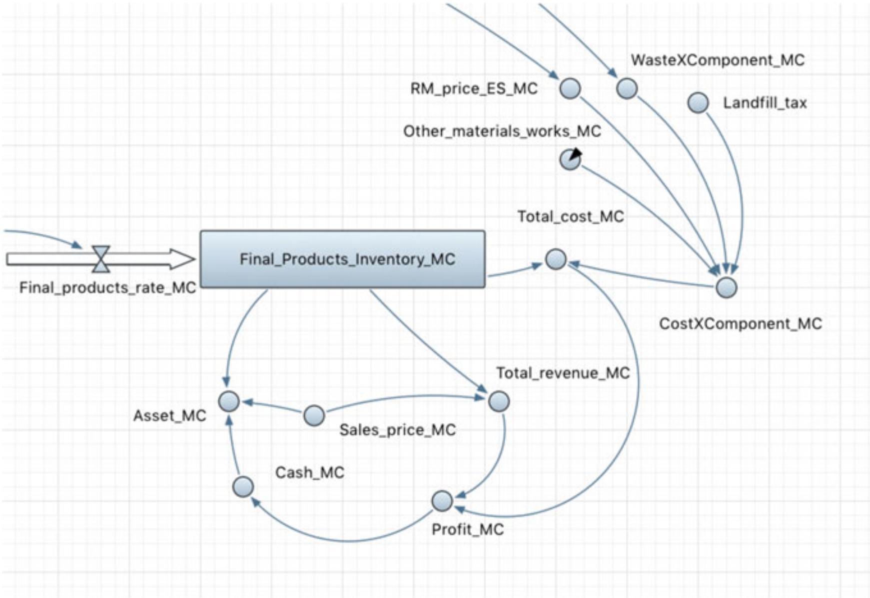


Fig. 3 Revenue, cost and profit detail of the hybrid model

terms of income and outflow money, while asset analyses the value of the company in terms of profit, cash and value of warehouses.

On the contrary, if there are symbiosis linkages between firms, total costs result by summing the purchasing cost, which now depends on the cost policy adopted by the symbiotic firm, and costs of other material and works. Clearly, disposal costs are not mentioned anymore. At this point it is important to underline that in some case waste are effectively “sold” as iron cast which maintains a value, but in other cases the company, which wants to sell its waste, must pay another other firm for taking its waste. The firm will be able to pay at most the landfill tax. In this case revenue is given by the final product sold and the profit derived from this symbiosis.

Finally the whole mathematical formulation is:

$$Production\ rate = min(Production\_capacity\_MC,\ Raw\_Material\_Inventory\_MC)/Production\_time\_MC \quad (1)$$

$$Waste\ rate = 0.05 * Service\_inventory_{MC} \quad (2)$$

$$Final\ produc\ rate = 0.95 * Service\_inventory_{MC} \quad (3)$$

$$CostXComponent_{MC} = Landfill\_tax * WasteXComponent_{MC} + RM\_price\_ES_{MC} + Other\_materials\_works_{MC} \quad (4)$$

$$Total\ cost = (CostXComponent_{MC} * Final\_Products\_Inventory_{MC}) \tag{5}$$

$$Total\ revenue = (FinalProductsInventory_{MC} * Salesprice_{MC} \tag{6}$$

$$Profit = Total\ revenue - Total\ cost \tag{7}$$

$$Cash = Initial\ value + Profit \tag{8}$$

The ABM model was described to explain the hybrid approach; the model consists of core entities called agents and links. Figure 4 shows in detail the ABM-SD approach.

Each firm within the industrial network is modelled as an agent, which decides to establish or not a symbiotic relationship with another firm belonging to the feasible industry. Hence, four agents named as the aforementioned firms compose the model; it has been assumed that each agent has a population of 100 firms. A symbiotic function is defined, which measures the willingness of a firm i to exchange wastes with a firm j and vice versa. Shown below, the symbiotic function is calculated considering that the firm j is selling its waste to the firm i. At this point, it is important to underline that the symbiotic function takes in account also

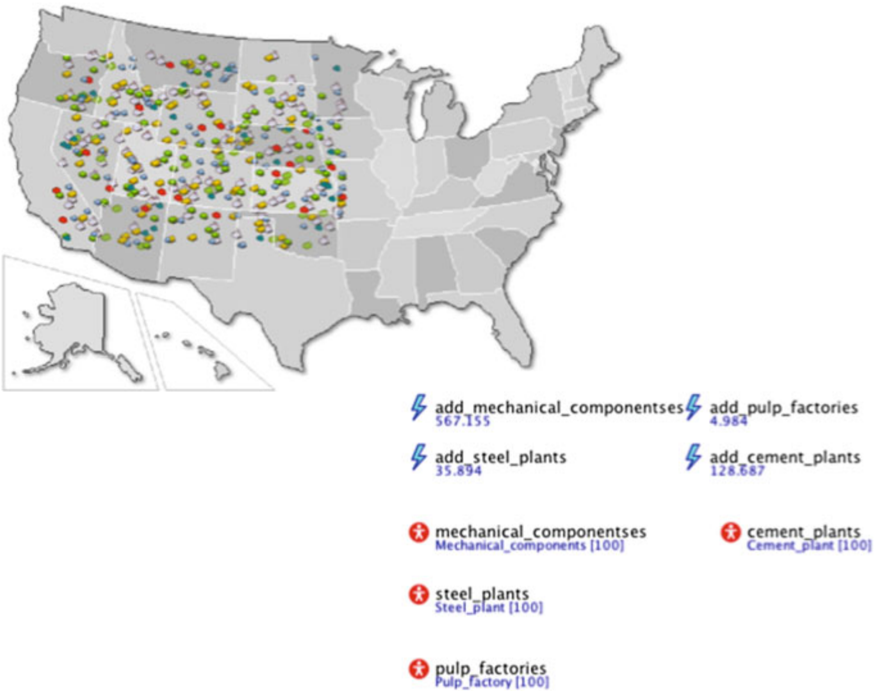


Fig. 4 Detail of the ABM-SD approach

pre-processing costs, in order to make external supplier raw material price competitive with the waste price.

$$\begin{aligned} \textit{Symbiotic function}\ (\mathrm{i} \rightarrow \mathrm{j}) = \textit{Money from}\ (j \rightarrow i) - \textit{Preprocessing cost}\ (i) \\ + \textit{External supplier raw material price} > 0 \end{aligned} \tag{9}$$

$$\textit{Symbiotic function}\ (\mathrm{j} \rightarrow \mathrm{i}) = \textit{Money from}\ (j \rightarrow i) \leq \textit{Landfill tax} \tag{10}$$

The abovementioned symbiotic function will be calculated in the state chart of each agent, the required data being taken from the SD model. The state chart of each agent is composed by two states: “No symbiosis” and “Symbiosis”. To move from one state to another, it is necessary that the symbiotic condition is verified. Thanks to the agents’ network, another important variable has been introduced—the Green Image Factor (GIF). In fact, IS has a positive impact on the company’s image. Therefore, it has been supposed that if a company has symbiotic linkage, it will reach an increase of sales; companies can also afford higher prices. Also in this case, GIF is calculated thanks to the introduction of an event “Calculate GIF”, which takes data from the SD model. Furthermore, the ABM model allows implementing another important aspect of the firm life cycle: birth and death. In fact, accordingly with the profit generated by firms and visible in the SD model, new firms/agents are pushed to enter in the market while other ones must leave the market caused by high costs.

Concluding, as previously discussed, the SD model allows considering flows and feedback dynamics from an aggregated viewpoint, while the ABM model allows describing in a clearer manner the behaviours within the company.

3 Results and Discussion

In this section, results obtained from simulation runs are reported and discussed following the two research questions. Table 6 reports simulation parameters.

3.1 Which are the Individual Behavioural Aspects that May Favour Industrial Symbiosis?

Figure 5 shows simulation results for the raw material scenario. We find that there is a correspondence between the raw material demand, price and number of IS. In fact, if external supplier’s demand decreases because of the increasing number of IS, there is a resulting decrease in raw material’s price. This means that an external supplier needs to reduce its price in order to be competitive with waste prices. This behaviour causes IS to be not a convenient approach in long term. We notice that

Table 6 Simulation's parameters

	Simulation parameters
Model unit time	• Week
Landfill tax	• 18 €/ton
Input purchasing cost MC	• 1800 €/ton
Sales price MC	• 208 €/Pulley
Input purchasing cost SP	• 250 €/ton
Sales price SP	• 1200 €/ton
Input purchasing cost CP	20 €/ton
Sales price CP	120 €/ton
Input purchasing cost PF	85 €/ton
Sales price PF	2000 €/ton
Waste cast iron price	• 140 €/ton

when raw materials' prices decrease, these causes a slow reduction of the number of IS too.

Another interesting behavioural aspect is linked to pre-processing cost (Fig. 6). When the number of IS decreases, pre-processing costs are high; differently, when the number of IS increases, a decrease results in the pre-processing cost. This behaviour is more clear if at the start of simulation there is a low number of IS, that is only few firms are able to provide pre-processing production. Conversely, when IS increases the pre-processing market becomes more competitive, as firms acquired knowledge.

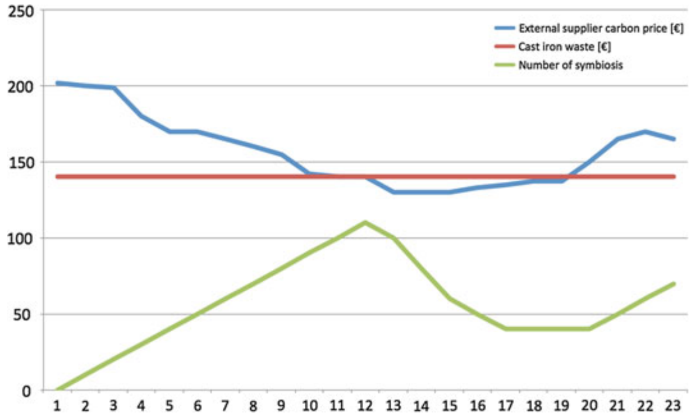


Fig. 5 Raw material's price behaviour with respect to the number of IS

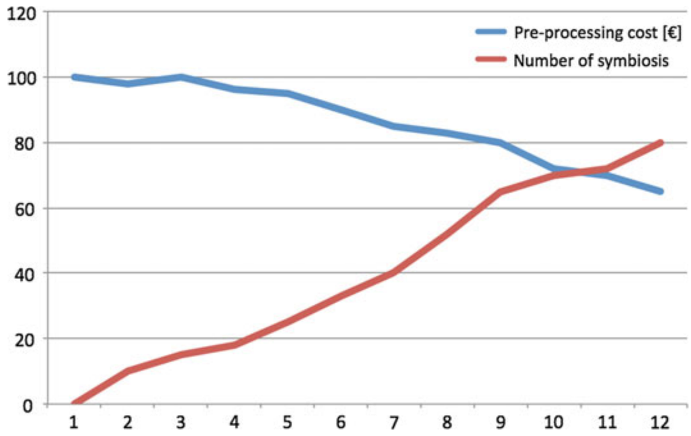


Fig. 6 Pre-processing's cost behaviour with respect to the number of IS

3.2 *Which are the Overall Benefits Associated with the Entire Industrial Network?*

Finally a comparison between the scenario with no symbiosis and the symbiotic one is reported (Figs. 7, 8, 9 and 10). We notice that the difference between the two profit is small at the beginning of the IS, and then increases over time. This aspect could be caused by: (i) complex interaction of technologies and process, (ii) technical and regulatory barriers, which can be overcame over time.

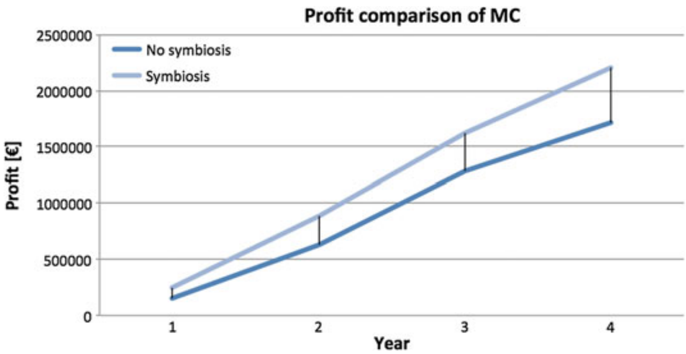


Fig. 7 Profit comparisons for MC

4 Conclusions and Future Developments

In the current state of development of the model, the results are promising but the model still needs further revision, for improvements and enhanced validation in order to deliver more realistic results. However, its design is seen as an approach to modeling multi-agent network systems that may serve as the basis for the development and sustainability of industrial symbiosis.

One of the limitations of the present work is that transportation costs are not addressed. Even though, the current model is a good foundation for further iterations, and could be a good starting point to better investigate the hybrid simulation field.

Appendix

See Figs. 8, 9 and 10.

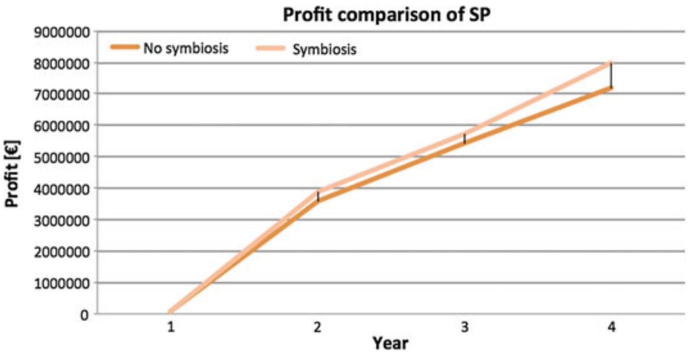


Fig. 8 Profit comparisons for SP

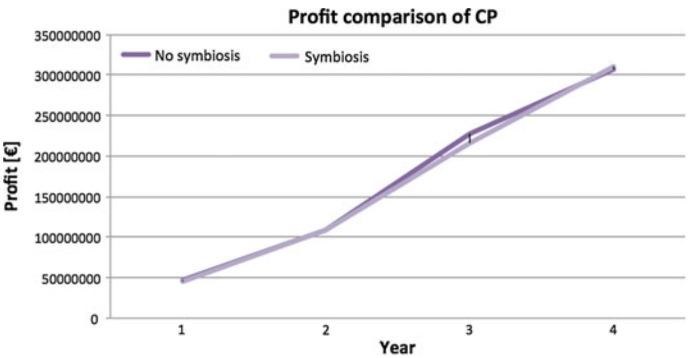


Fig. 9 Profit comparisons for CP

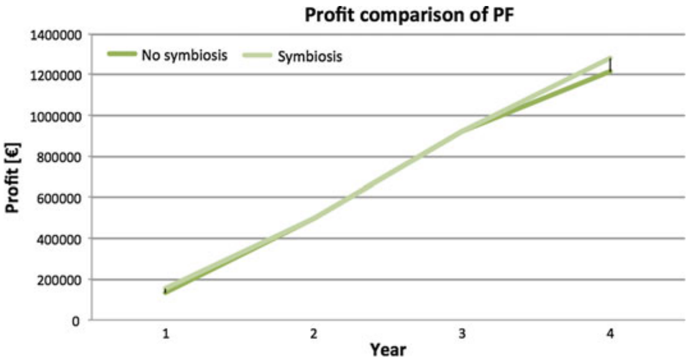


Fig. 10 Profit comparisons for PF

References

1. Chertow, M.R, Park, J.: Scholarship and Practice in Industrial Symbiosis: 1989–2014. In: Taking Stock of Industrial Ecology, vol. 1, pp. 1–362 (2016). https://doi.org/10.1007/978-3-319-20571-7_5
2. Tonelli, F., Paolucci, M., Demartini, M., Anghinolfi, D.: Multi-agent framework for manufacturing sustainability analysis and optimization. In: Proceedings of the SOHOMA’16 Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing. Springer (2016)
3. Lee, S., Geum, Y., Lee, H., Park, Y.: Dynamic and multidimensional measurement of product-service system (PSS) sustainability: A triple bottom line (TBL)-based system dynamics approach. J. Clean. Prod. **32**, 173–182 (2012). https://doi.org/10.1016/j.jclepro.2012.03.032
4. Golroudbary, S.R., Zahraee, S.M.: System dynamics model for optimizing the recycling and collection of waste material in a closed-loop supply chain. Simul. Model. Pract. Theory **53**, 88–102 (2015). https://doi.org/10.1016/j.simpat.2015.02.001
5. Abdelghany, M., Eltawil, A.B.: Linking approaches for multi-methods simulation in healthcare systems planning and management. Int. J. Ind. Syst. Eng. **X**(2), 1–16. (2017). https://doi.org/10.1504/IJISE.2017.083676
6. Lättilä, L., Hilletofth, P., Lin, B.: Hybrid simulation models—When, why, how? Expert Syst. Appl. **37**(12), 7969–7975 (2010). https://doi.org/10.1016/j.eswa.2010.04.039
7. Monasterolo, I., Jones, A., Tonelli, F., Natalini, D.: A hybrid system dynamics—agent based model to simulate complex adaptive systems : a new methodological framework for sustainability analysis. In: Proceedings of the 32nd Internationl Conference of the System Dynamics Society June 2016, pp. 1–18 (2014)
8. Wang, B., Moon, Y.B.: Hybrid modeling and simulation for innovation deployment strategies. Ind. Manag. Data Syst. **113**(1), 136–154 (2013). https://doi.org/10.1108/02635571311289719
9. Masoudipour, E., Amirian, H., Sahraeian, R.: A novel closed-loop supply chain based on the quality of returned products. J. Clean. Prod. **151**, 344–355 (2017). https://doi.org/10.1016/j.jclepro.2017.03.067
10. Tonelli, F., Fadiran, G., Raberto, M., Cincotti, S.: Approaching industrial sustainability investments in resource efficiency through agent-based simulation. In: Springer Studies in Computational Intelligence, vol. 640 (2016). https://doi.org/10.1007/978-3-319-30337-6_14
11. Xu, Z.X., Takeuchi, K., Ishidaira, H., Zhang, X.W.: Sustainability analysis for Yellow River water resources using the system dynamics approach. Water Resour. Manage. **16**(3), 239–261 (2002). https://doi.org/10.1023/A:1020206826669

How to Build a "Cooperative" Safety Bubble for a Reconfigurable Assembly System?

Yves Sallez and Thierry Berger

Abstract Reconfigurable manufacturing systems have been proposed in the last decade to deal with mass-customization problems and volatile market environment. If physical design, control or scheduling issues of these systems have been studied intensively, very few works concern the inherent safety problems. However this issue is of first interest for reconfigurable robotized cells with frequent interventions of human operators. The present paper must be considered as a position paper introducing the new concept of "cooperative" safety bubble. This last aims to insure operator's safety by cooperation between "safe" robotized units and safety devices. A preliminary implementation methodology of such safety bubble is presented and some corresponding research issues are proposed.

Keywords Reconfigurable manufacturing system · Reconfigurable assembly system · Collaborative robotics · Safety

1 Introduction

Nowadays, industry must deal with the fluctuation of markets and increasing product variants involving important fluctuations in production volumes. Manufacturing systems must continually and efficiently adapt their production. In this context, the concept of Reconfigurable manufacturing systems (RMS) was introduced by Koren et al. [1] to increase the speed of responsiveness of manufacturing systems to unpredicted events (e.g., sudden market demand changes, machine failures). The goal is to offer exactly the capacity and functionality needed, when required. RMS can be defined as follows: "Reconfigurable Manufacturing Systems are designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and

Y. Sallez (✉) · T. Berger
University of Valenciennes and Hainaut-Cambrésis,
LAMIH UMR CNRS n°8201, F-59313 Valenciennes, France
e-mail: yves.sallez@univ-valenciennes.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_14

functionality within a part family in response to sudden changes in market or regulatory requirements" [2]. RMS constitutes a new class of production system that lies between dedicated lines and flexible manufacturing systems [1].

Applied to the domain of the assembly, this concept gives birth to *Reconfigurable Assembly System* (RAS) [3–5].

In this context, this paper deals with a RAS demonstrator currently developed in our team by coupling of mobile robotized units. In this innovative type of production system, the operator safety is an important issue to consider. The paper focuses on this problem and is organized as follows.

In the first section, our motivations are presented. After a brief presentation of the RAS context, the demonstrator currently developed in the LAMIH Lab is detailed and safety considerations are highlighted. The second section is dedicated to the proposition of two ways (off-line and on-line) to manage safety problems in a RAS. The concept of "cooperative" safety bubble and an implementation methodology are equally presented. In the third section, in relation with the previous concept, several topics are proposed for future research works. Finally, conclusion is offered in the last section.

2 Motivations

As seen previously, RAS consists of hardware modules (e.g., robots, conveyors) and associated software components that can be added or removed according to the capacity requirement [3, 4]. An overview of RAS and design guidelines is available in [5, 6] and several works and projects are cited below:

- One of the more advanced prototypes is the factory system developed in the University of Windsor [3]. All modules are equipped with optical devices allowing the control system to recognize the topology of the cell. Work pieces are transported within each module using two levels conveyor.
- In the context of the hyper flexible automatic assembly project (HFAA), Onori and Alsterman [7] proposed a solution, where manual assembly stations can be extended gradually by standardized automated assembly modules.
- Chen [8] identified the importance of 'plug and play' modules and highlighted their use for reconfigurable robotized work cells. Modular building blocks were used to build parallel robots.
- In [9] the authors introduce mobile and modular assembly modules to build a RAS within final assembly lines of an automobile manufacturer. Major changes of the assembly unit can be obtained with little effort and short time. The mobile assembly modules have to ensure the possibility of the collaboration of humans and robots.

Currently, new technologies like collaborative robots allowing human-robot cooperation could facilitate the development of RAS. In [10–12] the authors offer a survey on collaborative robotics and compare manual operations, traditional robot cells and collaborative installations. However, if some works [5, 6, 13] explore the design, control or scheduling of RAS, very few deal with human-robot collaboration and the inherent safety issues [9, 14].

In addition, safety must also be considered according to PHM (Prognostics and Health Management) point of view. In [15], the authors argue that if a robotic system experiences a failure, it is expected to do so in a safe, reliable manner that does not negatively impact its environment, process, or collaborators. In this PHM context, [16] proposes a methodology to identify the positional health (position and orientation accuracy) changes.

Our team is actually working on the development of a RAS demonstrator in the field of electronics industry for the assembly of electronic boxes. While the robots perform components assembly and screwing tasks, the operators carry out more delicate tasks like cabling or very precise insertion. This RAS is built by coupling mobile robotized units (moved by the operators) and physically attached ones via specific fixtures. As depicted Fig. 1, the layout containing mobile units, tables and

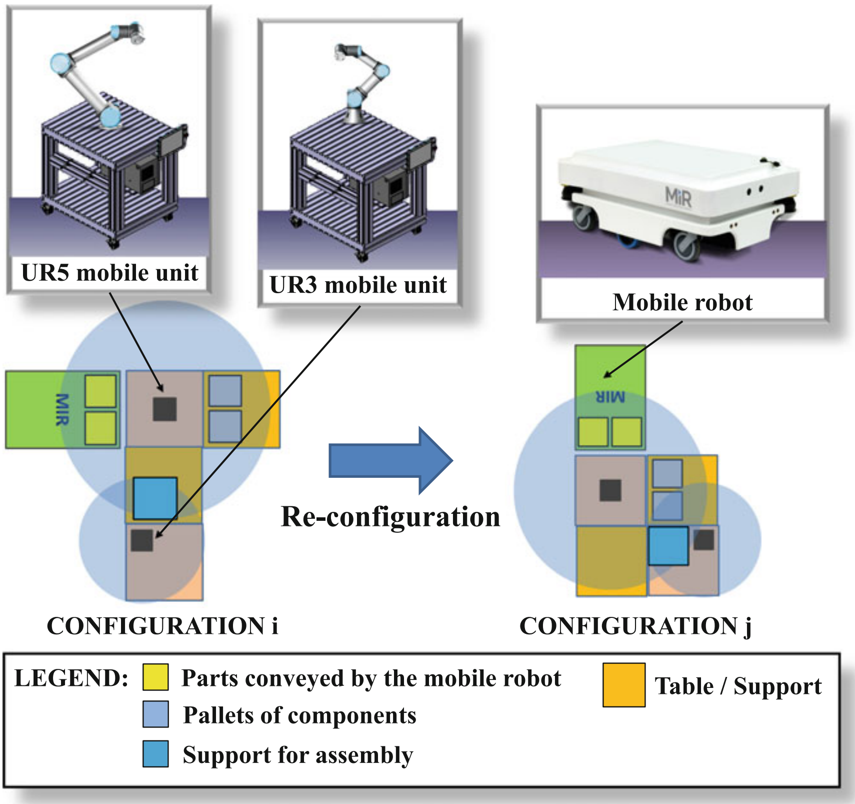


Fig. 1 RAS currently in development in LAMIH

supports corresponds to a "stationary" configuration of the RAS. This configuration remains valid until a new re-configuration.

In addition, some mobile devices (i.e. mobile robots conveying parts or components pallets) can temporary enter in the robots workspaces. Depending of the required assembly tasks, the operators intervene according two modes: occasionally during limited time for some specific tasks (e.g., feeders reloading, short maintenance task…) or for a long time to participate at collaborative works with the robots.

From safety point of view, this type of cell is situated between two extremes. The first one corresponding to a classical robotized cell with minimal interaction with operators, and the second one corresponding to a fully collaborative robotized cell where the operator works continuously with the robot. The first case is well mastered in industry and classical protective solutions (e.g., fences, infrared barriers, laser radars…) are used to detect operator intrusion in the robot workspace. The second case is intensively studied through numerous projects [17–20]. The safe "physical" interaction with robots is studied by the "physical Human–Robot Interaction" community [21, 22]. Nevertheless, the workspace being common with the operator, the speed of the collaborative robot is limited by regulation to reduce the risks. This speed limitation can induce negative repercussions on the productivity.

In our case, the aim is to use the robots at high speed when the operator is outside the RAS and to adapt the speed or stop the robot when the operator enters in the workspace. The mobility and the re-configurability of the robotized units induce new safety constraints. As shown Fig. 2, if some RAS demonstrators [3, 23] solve the problem by a full coverage of the robotized modules, we argue that new technologies offer more "open" solutions allowing co-existence of mobile robotized units, mobile robots and collaborative/manual workstations. These new solutions are explored in the next sections.

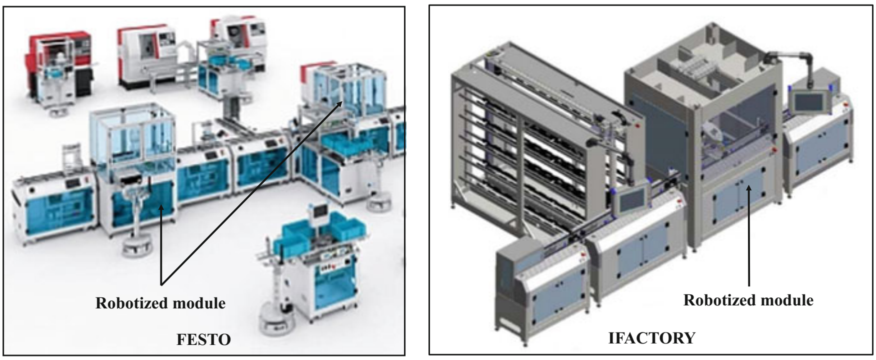


Fig. 2 Full coverage of robotized modules [3, 23]

3 Towards a "Cooperative" Safety Bubble

3.1 Methodology to Address the Safety Problems

In our view, two types of approaches can be proposed to face safety problems in RAS. The first approach, denoted "off-line", is relatively classical and lies on an analysis of a limited number of configurations.

After an analysis of the robotized tasks to perform, a list of possible configurations is built. Each configuration is relative to a layout of the RAS with a specific location of the different entities (e.g., robots, tables, supports…). As depicted Fig. 3a, a safety analysis is performed for each configuration.

Different methods are available to support this analysis [24–27]. As an example, Dhillon [24] presents robot safety-related facts and figures along with the most useful seven safety analysis methods and the application of the two most widely used methods (i.e., fault tree analysis and Markov analysis). Another example, the IDAR method proposed by CETIM [27] considers the following steps: identification of the applicable directives, risk analysis, identification and processing of safety functions.

If this "off-line" approach is classically well mastered, it can be tedious to analyse all the possible configurations and certify each of them. In fact, only a limited number of configurations can be certified as "safe" and deployed in-situ.

The second innovative approach aims to build "on-line" a "cooperative" safety bubble (see Fig. 3b). The aim of this approach is to take into account dynamically

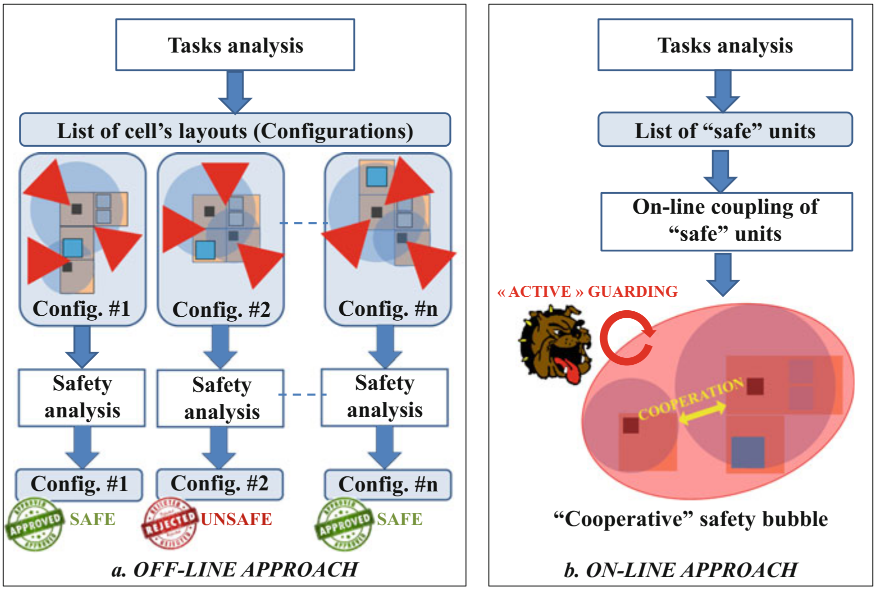


Fig. 3 "Off-line" and "On-line" approaches

the safety problem allowing a more easy re-configurability of the cell, not restricted to a limited number of "certified" configurations. This approach lies on a coupling of units able to take into account cooperatively the problem of safety. Each unit is assumed to be "safe", able to detect the presence of operator entering in its workspace. The cell's layout obtained by coupling of these units exploits the cooperation among units to insure the safety of the set.

Safety rules (e.g., distance to respect between human and robot in case of high speed, in any circumstance a possibility for the human to exit the cell…) have to be applied to check the validity of the layout. When the RAS is operational, these rules are used on-line by the "active" guarding in monitoring the safety of the cell.

If this approach seems attractive for its easiness of deployment, some issues must be solved to implement in-situ the proposed concept.

3.2 The Concept of "Cooperative" Safety Bubble

This concept is based on the cooperative behaviour of the different entities (i.e., robots, sensors…) composing the reconfigurable cell. The "cooperative" safety bubble exploits the detection capabilities of each "safe" unit and the interactions (concerning the safety) among units. Intrinsically each "safe" unit is assumed to manage its own safety by means of some specific devices (e.g., fences, laser scanner). When a unit detects a human intrusion, it sends the information to the

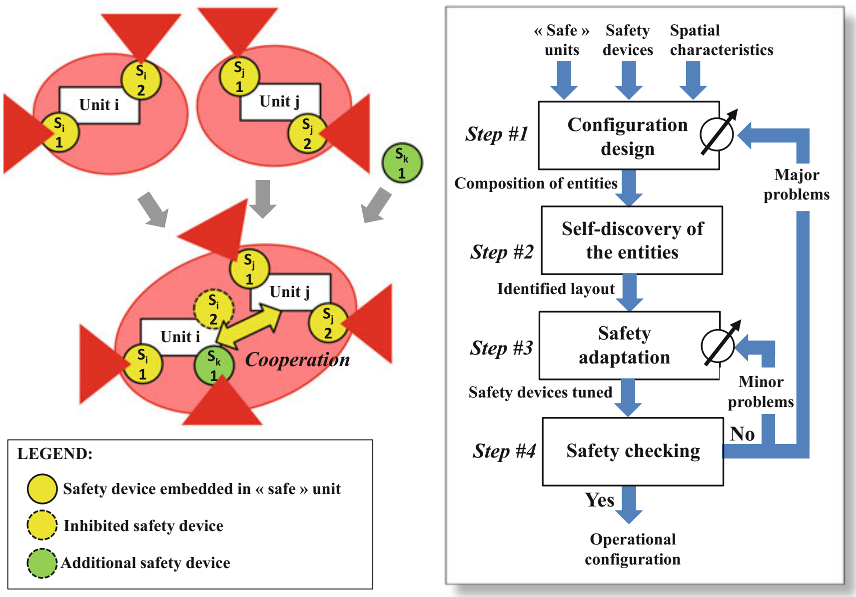


Fig. 4 Methodology to build a "cooperative" safety bubble

other units that can adapt their behaviours (e.g., speed reduction, halt). Figure 4 (right) depicts the proposed methodology to build a “cooperative” safety bubble.

The first step corresponds to the reconfiguration of the cell. The “safe” units are simply located one next to the other and coupled via specific fixtures. The safety devices associated to each “safe” unit remain attached to the unit. If necessary, other safety devices (e.g., fences, 3D camera) can be added to the layout to totally satisfy the safety rules.

All entities are assumed to be coupled physically and informational (i.e. they are equipped with communication devices to share safety information). The spatial characteristics of the shop-floor (where the configuration is installed) must be equally considered. For example, the presence of walls or pillars can complicate or simplify the safety considerations, according the different cell’s layouts. The output of this step is a composition of safe entities (i.e. a configuration to check in regard of safety).

The second step is dedicated to the self-discovery of the different entities (i.e., safe units with their safety devices, other added safety devices) composing the cell. These entities interact with one another to identify their respective locations in the new layout. The result of this step is the topology of the cell layout.

The third step consists in an adaptation (or tuning) of the different identified safety devices. Depending of the previous identified topology, some safety devices must be totally inhibited or tuned to avoid interferences among the units (see Fig. 4 left). The additional safety devices must equally be tuned. The spatial characteristics of the shop-floor, the operators’ locations and their pathways must be taken into account during this tuning phase.

The fourth step is crucial to insure the safety of the candidate configuration. A safety checking (based on the safety rules) is performed to detect any safety flaw allowing an operator to enter in the cell without detection. This checking exploits the layout topology, the performance characteristics and locations of the different safety devices.

At the end of this step, if the configuration is considered as “safe”, the new one can become operational. This safety checking can be performed automatically or with the intervention of a human safety expert who could “certificate” the configuration.

If the result of the safety checking is not satisfying, a return to the adaptation step#3 is performed. If the tuning seems too tedious to realize, the process must be redone from step#1.

Once the safety has been validated, the cell can become operational. During the exploitation phase, an “active” safety guarding must be insured by interactions among entities. The obtained “cooperative” safety bubble must react in “real-time” to any human intrusion in the robots workspaces. Depending of the location of the intrusion, some robotized units are halted and some others reduce their speed.

In this context, the next section is dedicated to a proposal of some research topics.

4 Issues to Create a "Cooperative" Safety Bubble

To build an efficient "cooperative" safety bubble, several important issues must be considered:

Issue#1: Which safety rules must be applied for an efficient safety checking of a "stationary" configuration? This last corresponds to the coupling of the "safe" units and the optional adding of safety devices, without the presence of temporary entities such as mobile robots, trolleys… This issue includes some sub-problems:

- How to realize the self-discovery of the different "safe" units and safety devices located in the cell to be "sure" of their effective presence?
- How to determine the different configuration principles allowing:

– The inhibition of some safety devices to avoid safety "interference" between devices (e.g. two laser radars seeing each other)?
– The sharing and the cooperative usage of different safety devices (e.g., protective fences, programmable safety plans, laser radars, 3D sensors (see Fig. 5)) to insure the operator safety?

What methodology to apply to check the effectiveness of the safety bubble?

Issue#2: How to detect safely (i.e. always without error) human operators in a dynamic environment?

As seen previously, a specific configuration is composed of "permanent" entities and of temporary entities (e.g., mobile robots, operators). The safety bubble must distinguish the different types of temporary entities. This issue must lead to the

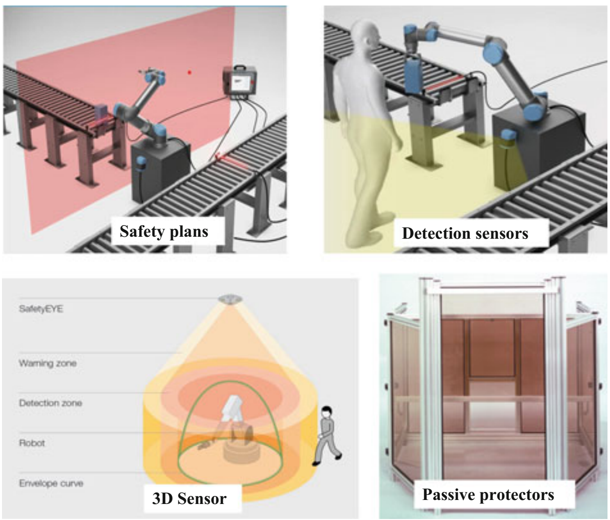


Fig. 5 Sharing and exploitation of safety devices

development of new sensors or algorithms able to analyse complex situations with human operators and mobile artificial entities.

This issue must lead to the development of new sensors or algorithms able to analyse complex situations with human operators and mobile artificial entities.

Issue#3: How design cooperative safety approaches using dissemination and fusion of information issued from different safety sensors?

For example, as illustrated Fig. 6, the information of the embedded lidars in the mobile robots can be used to detect the operator. When a mobile robot enters the detection field of the robotized unit#1, the detection capabilities of this last are partially inhibited (1) (and the robot#1 speed limited). However the embedded lidar detects the operator following the mobile robot (2). The mobile robot can then alert the unit#1 (3) to stop the robot. This type of cooperation illustrates the "active" guarding that must occur to insure an efficient "cooperative" safety bubble.

Issue#4: How to design a "sure" safety layer? The result of this issue is of first importance to implement really an "active" safety bubble. In fact, to be "sure" to obtain a safe RAS, the safety bubble architecture must react in "real-time" to any intrusion in the protected space. This implies to deal with different constraints of reactiveness, communication reliability and responsiveness.

This safety layer must equally deal with the heterogeneity of the different equipment (e.g., robot controllers, safety devices…) and interoperability between control units must be insured. In consequence, the robot controllers must be sufficiently open to integrate new functionalities (e.g., speed reduction according radar detection, definition of safety volumes).

Issue#5: Which tools and methodologies to assist the safety expert in the certification of the RAS configuration? Depending of the different countries' laws, it

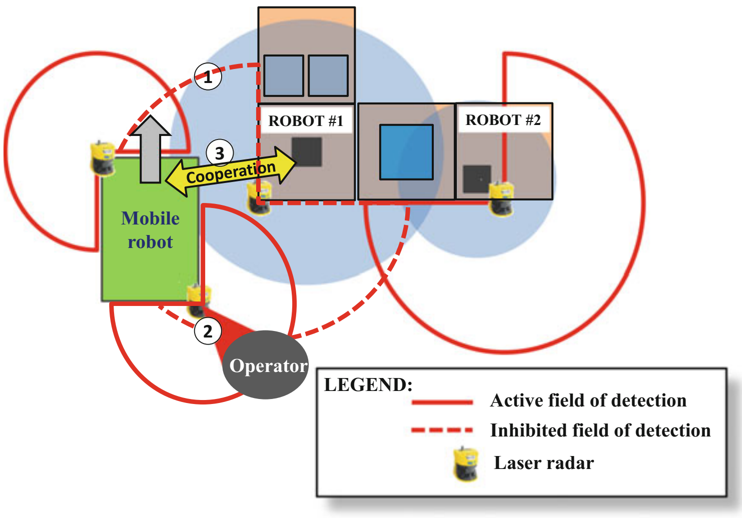


Fig. 6 Example of cooperation to insure a "cooperative" safety bubble

could be mandatory to obtain the validation of a human safety expert before to begin the production with the new configuration. In our point of view, this certification can lead to new functions among the operational staff. These tools could equally exploit some devices used in collaborative robotics to support the human-robot interaction (e.g. lighting devices representative of the robot status) or some augmented-reality tools.

5 Conclusion

This position paper has focused on RAS, based on modular structures able to deal with frequent changes in products and production volume and reduce time to market. The co-existence of mobile robotized units, mobiles robots and different supports induces safety risks for the operators that intervene in the cell. To face this safety issue, a new methodology based on the innovative concept of "cooperative" safety bubble has been proposed. This safety bubble exploits cooperation among robotized units and safety devices to detect any intrusion of the operators in the robots workspaces. This new concept induces several research topics addressing technical, methodological and organizational issues.

These topics will be studied in the next future on the RAS demonstrator currently developed in LAMIH. We hope that this position paper will instigate research works on the challenging issues proposed.

References

1. Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Van Brussel, H.: Reconfigurable manufacturing systems. CIRP Ann. Manuf. Technol. **48**, 527–540 (1999)
2. Koren, Y., Shpitalni, M.: Design of reconfigurable manufacturing systems. J. Manuf. Syst. **29**, 130–141 (2010)
3. ElMaraghy, H., ElMaraghy, W.: Smart adaptable assembly systems. Proced. CIRP **44**, 4–13 (2016)
4. Wiendahl, H.-P., ElMaraghy, H.A., Nyhuis, P., Zäh, M.F., Wiendahl, H.-H., Duffie, N., Brieke, M.: Changeable manufacturing-classification, design and operation. CIRP Ann. Manuf. Technol. **56**, 783–809 (2007)
5. Bi, Z., Wang, L., Lang, S.Y.: Current status of reconfigurable assembly systems. Int. J. Manuf. Res. **2**, 303–328 (2007)
6. Huettemann, G., Gaffry, C., Schmitt, R.H.: Adaptation of reconfigurable manufacturing systems for industrial assembly-review of flexibility paradigms. Concepts Outlook Proced. CIRP **52**, 112–117 (2016)
7. Onori, M., Alsterman, A.: Hyper flexible automatic assembly: needs and possibilities with standard assembly solutions. In: 3rd World Congress on Intelligent Manufacturing Processes & Systems (2000)
8. Chen, I.-M.: Rapid response manufacturing through a rapidly reconfigurable robotic workcell. Robot. Comput.-Integr. Manuf. **17**, 199–213 (2001)

9. Weber, J., Stäbler, M., Baumgartl, J., Paetzold, K.: Mobile assembly units as enabler for changeable assembly lines. Proced. CIRP **44**, 383–388 (2016)
10. Cherubini, A., Passama, R., Crosnier, A., Lasnier, A., Fraisse, P.: Collaborative manufacturing with physical human–robot interaction. Robot. Comput.-Integr. Manuf. **40**, 1–13 (2016). https://doi.org/10.1016/j.rcim.2015.12.007
11. Krüger, J., Lien, T.K., Verl, A.: Cooperation of human and machines in assembly lines. CIRP Ann. Manuf. Technol. **58**, 628–646 (2009)
12. Grahn, S., Langbeck, B., Johansen, K., Backman, B.: Potential advantages using large anthropomorphic robots in human-robot collaborative, hand guided assembly. Proced. CIRP **44**, 281–286 (2016)
13. Benkamoun, N., Huyet, A.-L., Kouiss, K.: Reconfigurable assembly system configuration design approaches for product change. In: Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM). IEEE, pp. 1–8 (2013)
14. Fechter, M., Foith-Förster, P., Pfeiffer, M.S., Bauernhansl, T.: Axiomatic design approach for human-robot collaboration in flexibly linked assembly layouts. Proced. CIRP **50**, 629–634 (2016)
15. Choo, B.Y., Beling, P.A., LaViers, A.E., Marvel, J.A., Weiss, B.A.: Adaptive multi-scale phm for robotic assembly processes. In: Proceedings of the Annual Conference of the Prognostics and Health Management Society (2015)
16. Qiao, G., Schlenoff, C., Weiss, B.A.: Quick Positional Health Assessment for Industrial Robot Prognostics and Health Management (PHM) (2017)
17. ROBO-PARTNER Project—ROBO-PARTNER Project Portal. http://www.robo-partner.eu/. Accessed 5 Sept 2017
18. EU Project: Factory-in-a-day. http://www.factory-in-a-day.eu/. Accessed 5 Sept 2017
19. rob@work. https://www.care-o-bot.de/en/rob-work.html. Accessed 5 Sept 2017
20. colrobot. https://www.colrobot.eu/. Accessed 5 Sept 2017
21. De Santis, A., Siciliano, B., De Luca, A., Bicchi, A.: An atlas of physical human–robot interaction. Mech. Mach. Theory **43**, 253–270 (2008)
22. Haddadin, S., Albu-Schäffer, A., Hirzinger, G.: Requirements for safe robots: Measurements, analysis and new insights. Int. J. Robot. Res. **28**, 1507–1527 (2009)
23. Industry 4.0—Training for the factory of the future—News—Festo Didactic. http://www.festo-didactic.com/int-en/news/industry-4.0-training-for-the-factory-of-the-future.htm. Accessed 5 Sept 2017
24. Dhillon, B.: Robot safety analysis methods. In: Proceedings of the 11th National Conference on Machines and Mechanics, pp. 86–93 (2003)
25. Do Hoang, Q.A., Guiochet, J., Powell, D., Kaâniche, M.: Human-robot interactions: Model-based risk analysis and safety case construction. In: Embedded Real Time Software and Systems (ERTS2 2012) (2012)
26. ISO, I.: ISO 10218-1: 2011: Robots and Robotic Devices–Safety Requirements for Industrial Robots–Part 1: Robots. International Organization for Standardization, Geneva, Switzerland (2011)
27. Dequaire-Falconnet, E., Meleton, L.: IDAR®: une méthode d'analyse des risques dans le cadre de la directive. In: Machines. CETIM (2013)

Part IV
Cloud- and Cyber-Physical Systems for Smart and Sustainable Manufacturing

Design of High Availability Manufacturing Resource Agents Using JADE Framework and Cloud Replication

Silviu Răileanu, Florin Daniel Anton, Theodor Borangiu and Silvia Anton

Abstract The paper proposes a methodology for replicating in the cloud software agents associated to the control of manufacturing resources. Replicating in Cloud Manufacturing Control architectures (CMfg) agents and their services results in a high availability (HA) decentralized control system. Agents' services and replicated data will be detailed in the paper. This methodology represents an extension of the generic agentification process which consists in associating a software agent to a physical entity in order to simplify the access to the resource's operations managed as services and easily accessed through standard messages in multi-agent control frameworks (MAS). The developed methodology is validated using the JADE framework. The paper explains how a JADE agent acts as intermediary between the MAS framework based on the exchange of standardized FIPA messages, and direct resource communication which is based on exchanging information over a TCP connection.

Keywords Multi-agent system · Private cloud · High availability Agentification

S. Răileanu (✉) · F. D. Anton · T. Borangiu · S. Anton
Department of Automation and Applied Informatics, University Politehnica of Bucharest, București, Romania
e-mail: silviu.raileanu@cimr.pub.ro

F. D. Anton
e-mail: florin.anton@cimr.pub.ro

T. Borangiu
e-mail: theodor.borangiu@cimr.pub.ro

S. Anton
e-mail: silvia.anton@cimr.pub.ro

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_15

1 Introduction

In order to become more efficient, current manufacturing control systems make use of the latest developments in ICT [1–3] such as embedded systems (local resource control), Wi-Fi (eliminate cables and easy access to remote resources), cloud (intensive computation and high availability), online identification systems (simplify the routing process), a.o. A major change generated by the introduction of these developments was the large scale usage of distributed control systems [4, 5]. Along with the clear advantages of decentralization (local control, easy access to resource data, directly monitor and gather process data, a.o.) [6–8] some critical challenges arise like managing increasing systems' complexity (decisions are taken through interactions rather than using centralized algorithms), compatibility (different resources have different interactions/communication protocols), scalability (adding more resource should not decrease the whole system's performance) and fault-tolerance [9]. The last challenge is of critical importance since some services that were traditionally offered by centralized powerful systems are now offered by embedded systems and they should offer the same service level (ex.: uptime, high availability) [10]. Consequently agent oriented frameworks (multi-agent system—MAS) or simply concurrent, fault-tolerant and scalable software are used on top of the physical decentralized control architectures to realize both the decisional and the communication/synchronization part [11–13]. These solutions provide mechanisms that address the fault tolerance and high availability issues (e.g.: JADE [14], Erlang [15], a.o.) but these have to be configured/adapted to the characteristics of the targeted process (interfaces/interaction protocols between control software and machine have to be developed).

As a consequence the paper focuses on the control part of discrete manufacturing systems trying to offer a solution for the fault-tolerance issue. The paper is structured in six sections: Sect. 2 describes the architecture of the agent attached to a resource and how it will be replicated to offer a continuous service in case of breakdown; Sect. 3 shows the implementation of the resource agent replication process; Sect. 4 provides an alternative for the virtual replicated agents solution; Sect. 5 presents a case study and experimental results and the last section is devoted to conclusions.

2 Fault Tolerant Structure of the Resource Agent

The current work continues the research in the domain of decentralized control of manufacturing systems using the multi-agent approach presented in [14]. It enriches the control solution with the replication of decisional agents representing resources in order to make the architecture fault-tolerant. The current implementation of the decentralized control system was done using Java Agent Development Framework (JADE) [16, 17]. The framework is specially designed for programming multi-agent systems according to FIPA standards for intelligent and physical agents (http://www.fipa.org): (a) standardized agent interaction through the FIPA Agent Communication Language (ACL) and (b) simple behaviour definition (cyclic, one shot, finite state machine, a.o.) used for decision making and data processing. Also, the fact that it is based on Java permits to run the associated agents on different platforms characterized by different operating systems (Windows, Linux, a.o.), operation modes (graphical or console) and power consumptions (desktop or embedded) which is the case of a manufacturing control system. Besides these two characteristics, JADE has additional features that exceed the FIPA requirements like architecture distribution, fault tolerance, persistent message delivery, graphical interfaces and a container-like structure [16].

2.1 Standard Architecture

The standard control architecture uses decisional agents attached both to resources and work-in-progress (WIP). These agents implemented in JADE and running on Java Virtual Machines (VM) are interfaces used to integrate products being executed and manufacturing resources into the control architecture (Fig. 1: products are assembled on pallets). According to [18] who realized a classification of intelligent devices used in manufacturing, the products being executed and the resources in the current architecture are: *Class 1 devices*: workstation (PC)-assisted shop-floor device for resources, and *Class 3 devices*: intelligent shop-floor device for products being executed. Class 1 devices (Fig. 1 on the right side) are integrated using the resource's associated workstation on which runs a JADE agent that communicates with the JADE platform containing all other agents representing the cell resources and products being executed. Class 3 devices (Fig. 1 on the left side) are directly integrated in the JADE platform using the Intelligent Embedded Device (IED) physically associated to the product on which a JADE agent also runs, in a similar way as for Class 1.

2.2 Fault-Tolerance Approaches

In theory there are two known approaches [19–21] for the implementation of a fault-tolerant, high availability pure software system: the first one, entitled *survivalist approach*, preserves functionality through replication of the agents while the second one, entitled *citizen approach*, relies on other agents which monitor execution and take corrective actions in case of failure. The main difference between the two approaches is that in the first case agents are designed to monitor their own state and handle exceptions while in the second case external entities monitor and supervise the fault-tolerant component. Moreover, the first approach based on replication techniques has two facets: active and passive replication. In the

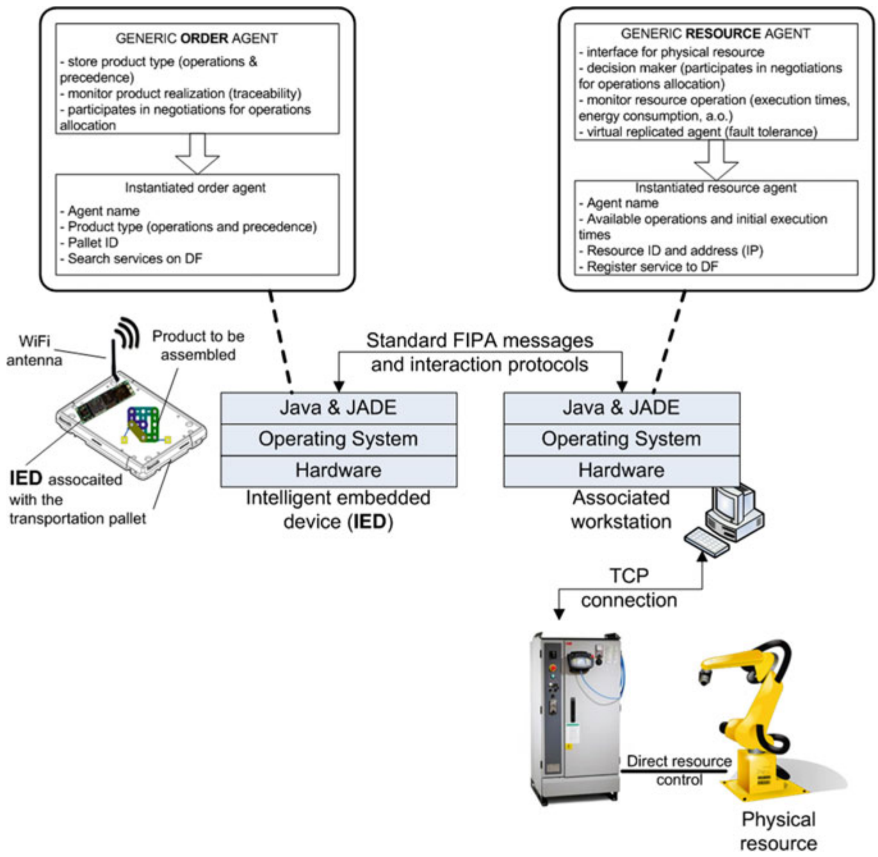


Fig. 1 Resource and order integration in the control architecture with JADE multi-agent frame

active replication case all existing replicas of the agent do the same operations, have synchronized data and the output is generated by a single entity. The survivalist approach, based on passive replication, maintains only a single active entity and the backup ones are activated only in case of fault; no information is provided about the internal state of the agent that will fail, except for the fact that the offered service continues to be provided.

2.3 *JADE Approach*

The JADE platform uses the survivalist replication approach to maintain the functionality of both the platform [16] and the composing agents [22]. Agents in JADE are structured and located in containers which are interconnected JAVA processes and can be distributed over a network. During platform initialization a *main container* is created which contains two critical agents for the MAS: Agent Management System (AMS) and Directory Facilitator (DF). Even if it is possible to have multiple platforms and send messages between them, all other containers must join the main container in order to have a single MAS platform. If the main container fails the AMS (which represents the MAS) and the DF shut down and consequently the MAS platform shuts down with all its connected containers. In order to avoid this scenario several backup main containers can be launched which actively monitor the master (initial main container). These are grouped in a ring of main containers and can use two types of policies for the master container monitoring: dynamic (Address-Notification service) and static (a fixed, known beforehand, list of main containers). Simple containers can be added to the ring of master and backup main containers and in these containers agents are located. If the master main container fails the AMS and DF agents are automatically moved to another backup main container assuring thus platform survivability.

But platform survivability is not sufficient; individual agents which offer services to other agents on a request/response base must also be fault proof. For this purpose JADE provides a mechanism entitled *virtual replicated agents*. As the name suggests this is a virtual agent (a name) and the platform takes care of directing the requests towards different replicas of an agent. The messages can be dispatched either to all available replicas or to a single replica. The virtual agent is a name, not an agent; it does not have a location while the replicas run on well-defined containers, preferably on different hosts (with different IPs) to assure survivability in case of failure. The platform takes care also of the synchronization between replicas' internal variables.

2.4 Problem Formulation

The manufacturing control system (Fig. 1) relies on the MAS framework for its operation and is thus susceptible to failures. We define a MAS failure as a known event that can happen at any time during the system's operation. The type of failure that we consider in our work is the unexpected death of an agent which can be caused by hardware failure (host shut down), software failure (error, exception or invalid input) or broken physical link in the network connection. The issue with this approach is that agents cannot execute directly on the resource controller due to warranty and security limitations and they exist in an intermediary layer (Fig. 1) acting as *middle man* between a FIPA environment (MAS) and a pure TCP environment (resource). This complicates the fault-tolerant problem which now shifts from an *agent fault-tolerant problem* to a *link fault-tolerant problem* (Fig. 2).

As stated in the "JADE approach" section for virtual replicated agents the MAS framework takes care of the synchronization of internal variables; however messages received by each agent are not distributed among all replicas. This situation was also verified and confirmed using the administrative and debugging tools offered by the JADE GUI: messages were sent between a service reader and several

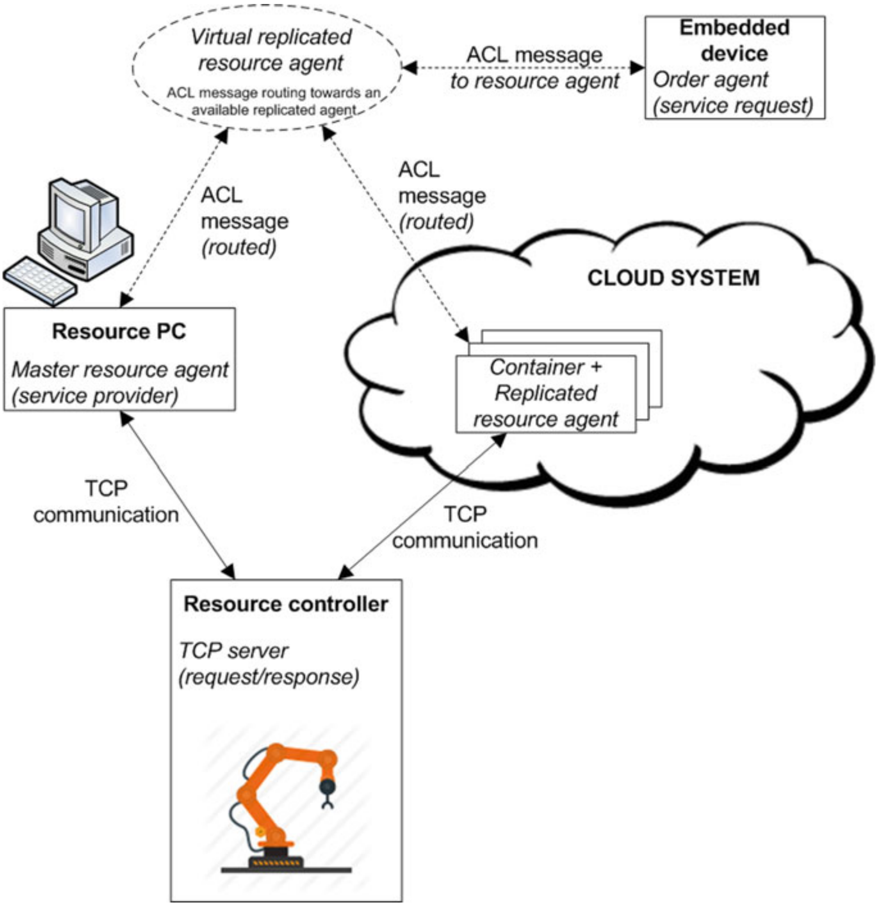


Fig. 2 The resource agentification model used for manufacturing control

service provider (replicated) agents; the messages have been visualized using the Sniffer agent and their queues were inspected using the Introspector agent [16]. As seen from the tests, messages routed to a given agent do not arrive at other agents if either HOT_REPLICATION (messages dispatched indifferently to all available replicas) or COLD_REPLICATION (all messages dispatched to the master replica only) is used.

Since fault-tolerant link towards the associated resource must be provided through a replicated agent, our solution is to combine the survivalist approach with the citizen approach: the resource agent is replicated, a single resource agent (master replica) makes contact with the resource and requests operations, and the state of the link along with all received requests are distributed among replicas as internal variables. Thus, in case of failure functionality is assured and the replica that takes over knows the state of the interaction protocol. The proposed interaction

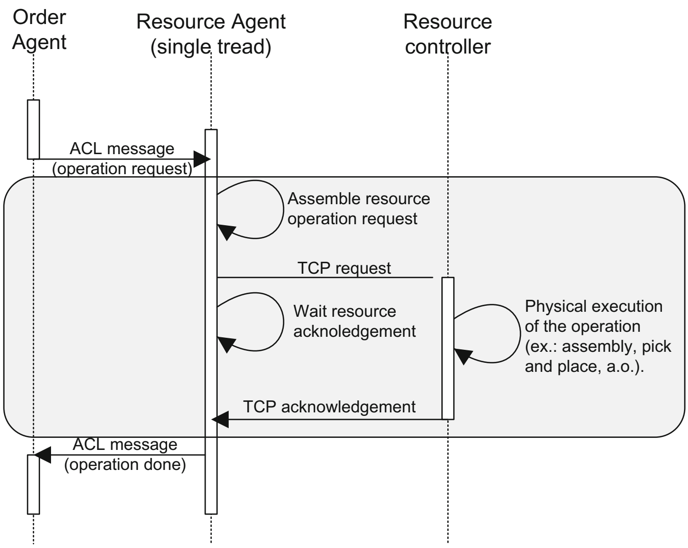


Fig. 3 Interaction protocol between the order agent, resource agent and resource controller

protocol and the possible failure cases appearance time is described in the sequence diagram below (Fig. 3).

The main problem consists in the fact that in the MAS framework the communication is asynchronous and based on agent names (ACL messages can arrive at any time and the agents are referenced by name not by the host they reside on), while between the resource agent and the resource communication is synchronous and IP-based (both the agent and the resource need to know their IPs in order to exchange messages; TCP is chosen over UDP to have reliable communication and not lose important packets).

Since agents are single threads, the agent-to-resource controller communication behaviour blocks the agent-to-agent communication behaviour and thus messages received in the interval of operation execution are not processed. If a failure occurs at this point confirmation cannot be sent back: the resource agent that initiated the communication (master replica) does not exist, the IP probably is no longer available (host or network failure) and the TCP session is no longer valid. For this reason (imposed by the implementation constraints) the proposed fault-tolerant architecture will focus on the case where the resource agent fails in between accessing the resource controller (Fig. 4). The master replica agent is automatically chosen by the MAS framework.

In order to have a pure fault-tolerant architecture, the case where failures occur in between the *operation request* and *operation done* messages should comply with the sequence diagram described in Fig. 5. In this case the agent, which is used as a proxy to access the resource, should be tested beforehand by the resource controller

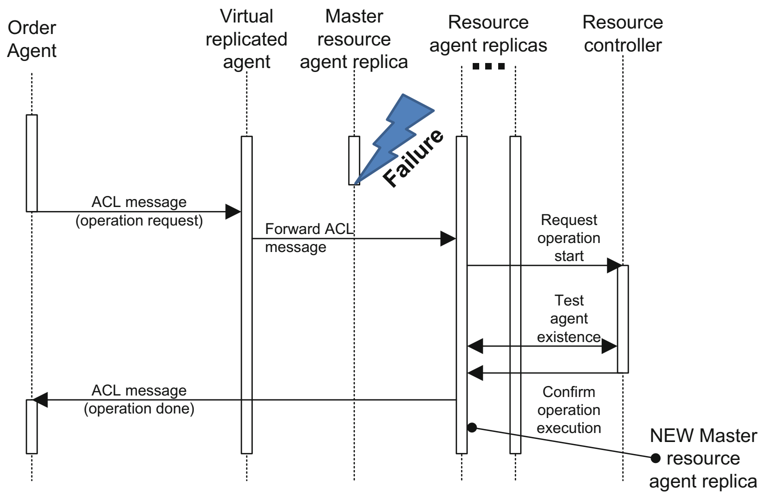


Fig. 4 Failure of the resource agent (Case 1: master resource replica fails (blue flash) before a request is issued by the order agent)

and only afterwards the confirmation message is sent. A solution which complies with this case is proposed in Sect. 3 (Alternative solution for high availability).

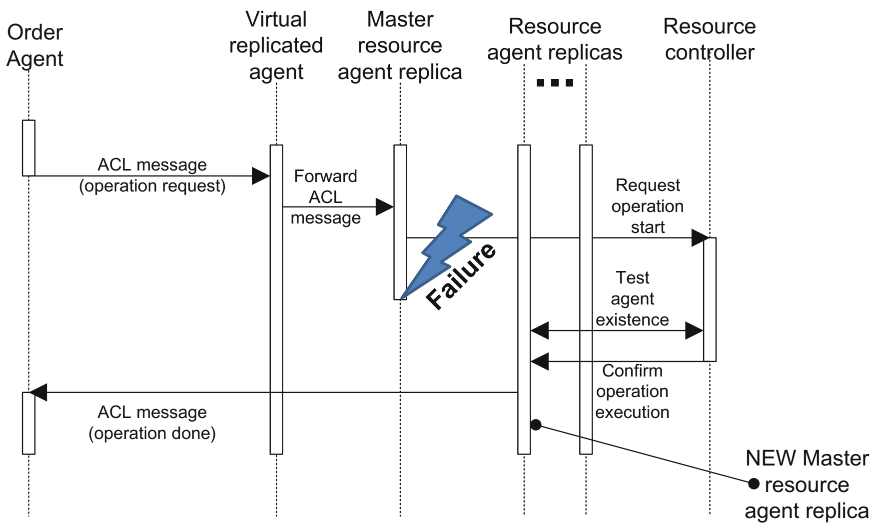


Fig. 5 Failure of the resource agent (Case 2: master resource replica fails (blue flash) after an operation request has been issued by the order agent)

2.5 Fault-Tolerant Architecture

The objective of this research is to extend a given architecture (Fig. 1) with fault-tolerant resource agents, meaning that they offer a continuous service to agents representing orders (high availability). Given the inherited constraints including the usage of the JADE framework, our proposition is to replicate these agents: instead of standard agents we extended them to virtual replicated agents (Fig. 2) and adapt them in order to suit the needs in accordance with the failure cases described before, Figs. 4 and 5. Thus instead of a simple agent executing continuously (no failure) a cyclic behaviour which translates ACL messages into TCP requests and vice versa, we now have an agent executing two threaded behaviours (a behaviour which receives ACL messages and a dedicated behaviour for TCP communication) with the associated replication code. The replication code consists of keeping replica agents in synchronization and handling replicas-related events like creation, removal, error processing and becoming a master. In our scenario (COLD_REPLICATION—messages sent only to the master replica) the minimal shared variables are the messages received by the master replica (ACLMessage[] ReceivedMessages) and their status which can be *waiting to be processed, in process* and *processed* (String[] ReceivedMessagesStatus) (Fig. 6).

Thus, the steps performed by a replicated agent are: (1) Read messages; (2) Update shared variables; (3) Send request to resources and wait for response (parallel thread); (4) Update shared variables; (5) Send confirmation message; (6) Test if at least one replica exists.

If no replica exists the operator is prompted to turn on a new host and launch a replica of the current agent on it. The existence of a replica is verified for each resource by polling the AMS for the existing agents and counting the ones associated to current resources. Each replicated resource agent has a fixed name (replica_X_resource_Y, X—an integer which is incremented with each replica,

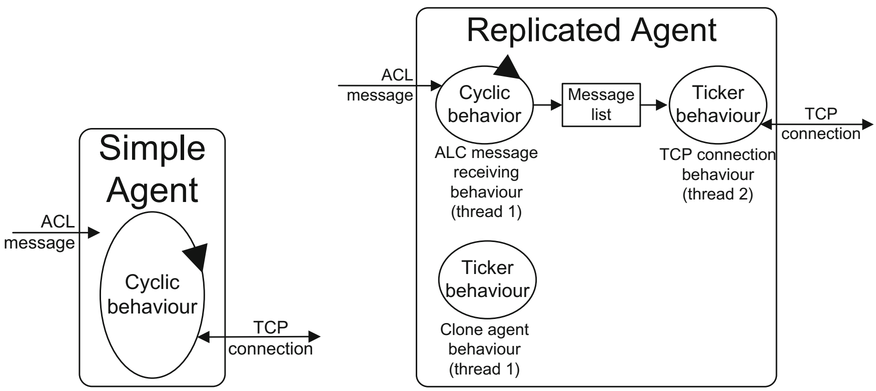


Fig. 6 Agent structure transformation (left—simple agent structure, right—replicated agent structure)

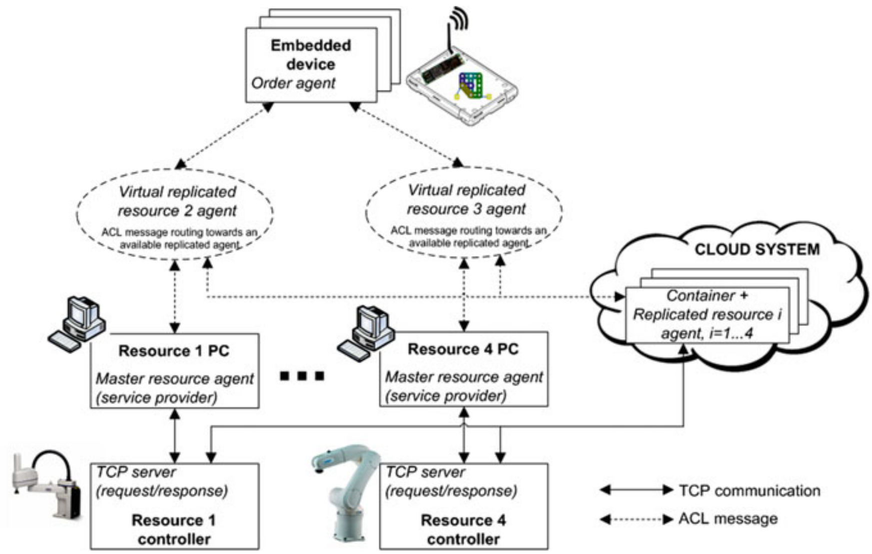


Fig. 7 Implementation of the fault-tolerant control system using HA resource agents

Y = 1, …, 4 (a maximum of 4 resources)). Initially there is a master replica on the PC attached to the resource controller and a replica running on the cloud on a dedicated host. As host fails new ones are created on the cloud. The complete implementation scenario of the high availability resource agents is presented in Fig. 7.

3 Alternative Solution to Provide High Availability

In order to offer high availability (HA) there are two solutions: (i) Offering HA at the level of *operating system* using virtual machines (VMs) or (ii) Using HA at *application level*. Both methods are trying to eliminate all single points of failure at hardware and software level. A single point of failure represents any hardware or software component of the system which, if is not working properly, the service which the system is offering becomes unavailable.

When offering a service using a HA cluster at the level of the operating system, the nodes in the cluster are physical machines of virtual machines which are configured in a HA cluster and one or more services are configured to run on these nodes [23].

A simple HA cluster configuration is depicted in Fig. 8. The nodes N1 and N2 are connected in two redundant networks: a service network which is used to offer the service to clients and an internal network used for monitoring and for cluster management. For cluster monitoring heartbeat packages are sent over the internal

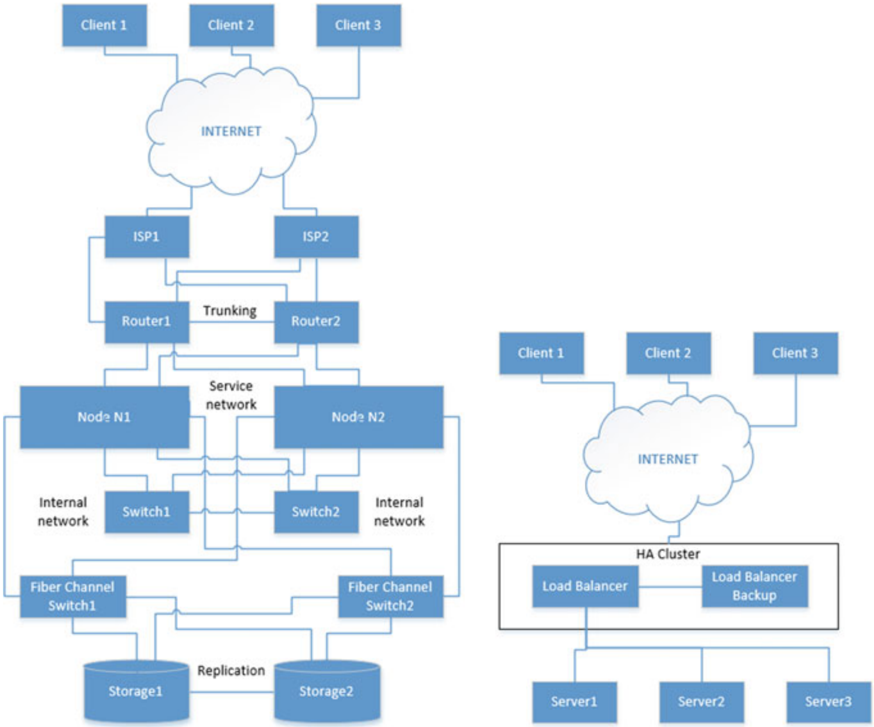


Fig. 8 The architecture of a HA cluster and a load balancer

network from node to node; when a heartbeat package is received the node answers with its status (available resources, and whether it is able to offer the services). In order to offer a service a node must have a set of resources associated, depending on the type of service which it is offering: IP address, the service itself (the application), a storage (accessible by all nodes in the cluster in order to offer the same service), network connectivity, etc.

If a resource is not available to the node, then the node will not be able to offer the service; in this case the heartbeat packages will detect the problem and the node will be isolated (or fenced) from the cluster, which means that the node will be either powered off or disconnected from the network and storage, and then the resources will be allocated to another node in the cluster and the service will be offered (relocated) by the other node [24].

The problem with the typical HA cluster is that a service runs only on one node, and that is limiting the cluster from the point of view of scalability. To solve this problem a load balancing system can be implemented (Fig. 8), in this case there is a number of nodes which are offering the same service, and in front of the servers there is a HA cluster which receives the requests from the clients and distributes them to the nodes according to a distribution algorithm.

This solution solves the scalability problem but has also a drawback, namely the resource's utilization. When using a hardware system to offer a service, the system will be used at about 30% of the capacity; this can be solved by virtualization which can increase the usage up to 85% [25, 26].

The presented solutions have also the advantage of isolating each service at the level of the node (physical machine or virtual machine), but sometimes they use too many resources. In order to obtain a better resource utilization and application isolation a solution using containers can be envisaged.

Now the HA facility can be implemented at operating system (OS) level or at application level (at the container level). The problem with running applications which are exchanging messages in HA clusters is that when a node becomes unavailable and the service is migrated the session is lost. Our approach is to offer HA with session preservation.

The proposed solution is based on Docker (http://www.docker.com): JADE agents are running in separate Docker containers and on separate VMs running on the cloud. JADE agents are implemented using Spring Session (http://spring.io/) which offers an easy way to replace a HTTP session in an application container, and also supports clustered sessions. Docker containers are clustered and managed in Swarm mode [27], which also offers load balancing, see Fig. 9.

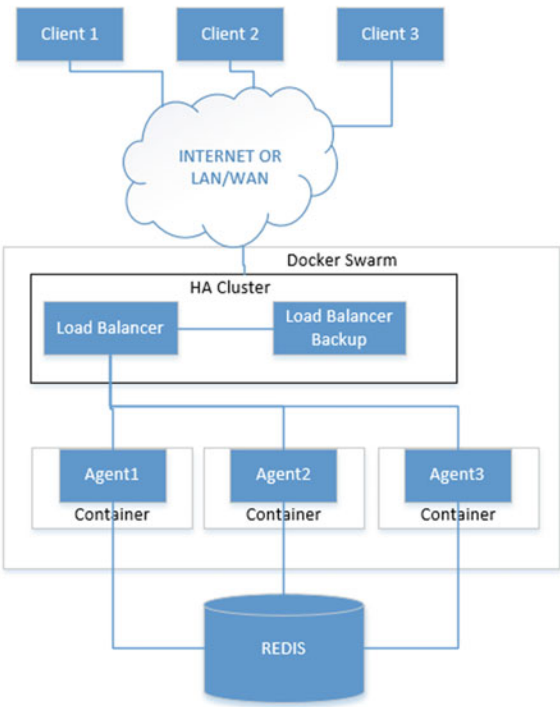


Fig. 9 The proposed architecture

In order to keep track of sessions and to preserve the sessions even if a container fails, the agents are using Redis (an open source in-memory data structure store, used as a database, cache, and message broker) which is appropriate for storing HTTP session information. Spring Session has default support for Redis [28], and it is very easy to make them work together.

From the point of view of cyber security, the system is composed by two local networks with restricted access through firewalls; these networks are connected using a VPN solution (OpenVPN). The first network hosts the production equipment and is completely isolated from attackers from internet because no services are allowed to be accessed (isolation is done at the firewall level). The equipment can communicate inside the network or with other services offered in cloud which are accessed through VPN. The second network is hosted in cloud using the same solution; because the connection is done over VPN, the data sent through Internet is encrypted.

4 Case Study. Experimental Results

Experiments have been carried out on the informational system for CMfg described in Fig. 7. These experiments have targeted both the functionality and the processing power used by the control architecture.

The experiments consisted in running a distributed JADE platform on the local computers (attached to manufacturing resources), intelligent devices (mobile, attached to orders) and cloud (for replication). During operation the resource master replica agent running on the computer was shut down. In this situation its functionality was shifted towards the replica located on the cloud. Concerning the agent replication function and the update of variables, the tests have been carried out according to specifications while replica switching is done instantaneous. The utilized processing power was measured by help of the Task Manager (Table 1).

Table 1 JADE platform RAM usage

<table>
<tr><th>Experiment</th><th>Memory usage by the JADE platform with services needed for replication, mobility and event notification</th></tr>
<tr><td>RAM used for launching MAS platform with a main container</td><td>Approx. 36.812 K</td></tr>
<tr><td rowspan="2">RAM used for adding the GUI to the main platform</td><td>Approx. 54.036 K</td></tr>
<tr><td>Main container process rises to 41,672</td></tr>
<tr><td rowspan="2">RAM used for adding locally a MAS container to the platform</td><td>Approx. 26.888 KK</td></tr>
<tr><td>Main container process rises to 42,076</td></tr>
<tr><td rowspan="2">RAM used for adding a MAS container to the platform (remote)</td><td>Approx. 23.2 K (win10)</td></tr>
<tr><td>Main container process rises to 43,208</td></tr>
<tr><td>RAM used for adding a master replica of the resource agent to the platform</td><td>Approx. 43.796 K (Main Container)</td></tr>
<tr><td>RAM used for adding a resource agent replica to the platform (to the 3rd row)</td><td>Approx. 29.848 K</td></tr>
</table>

Based on the realized experiments for the high availability solution the following observations are made: (i) the solution has a small footprint in utilization of the available resources, (ii) it is possible to offer a high availability service using only JADE when no physical equipment is connected but (iii) the solution is prone to fails when the link between the agent/resource controller breaks during operation reception.

5 Conclusions and Future Work

The paper describes an upgrade to an existing distributed manufacturing control solution based on the multi-agent technology. It focuses on the replication of resource agents on the cloud in order to provide them the capability of high availability service offered to order agents requesting operations. Different failure cases are considered and solutions are offered to each of them. The proposed upgrade solution works according to specifications in the case where failures occur when there is no communication between replicated agent and resource controller and the resource agent can handle multiple order agents in time.

Future research will focus on two problems: (i) the communication between the JADE agent and the resource controller and replacing the current TCP based interaction protocol with an UDP based interaction protocol; (ii) the usage of external solutions like Docker for high availability.

Acknowledgements This research work has been partially supported by the IBM FA 2016 project: Big Data, Analytics and Cloud for Digital Transformation on Manufacturing—DTM, period of execution 2016-2018.

References

1. Meyer, G., Framling, K., Holmstrom, J.: Intelligent Products: A survey, Computers in Industry. Elsevier (2008)
2. Trentesaux, D., Thomas, A.: Product-driven control: a state of the art and future trends. IFAC Proc. Vol. **45**(6), 716–721 (2012)
3. Sannapureddy, B.R.: Pros & Cons of IoT. https://www.linkedin.com/pulse/pros-cons-internet-things-iot-bhaskara-reddy-sannapureddy. Accessed June 2017
4. Peschke, J., Luder, A., Kuhnle, H.: The PABADIS'PROMISE architecture—a new approach for flexible manufacturing systems. In: IEEE Conference on Emerging Technologies and Factory Automation (2005). https://doi.org/10.1109/ETFA.2005.1612564
5. Zambrano Rey, G., Bonte, T., Prabhu, V., Trentesaux, D.: Reducing myopic behavior in FMS control: a semi-heterarchical simulation-optimization approach. Simul. Model. Pract. Theory **46**, 53–75 (2014)
6. Sahardis, G., Dallery, Y., Karaesmen, F.: RAIRO—Centralized versus decentralized production planning. In: Oper. Res. SMAI/ROADEF, **40**(2) (2006)

7. Babiceanu, R.F., Chen, F.F.: Distributed and centralised material handling scheduling: comparison and results of a simulation study. Robot. Comput.-Integr. Manuf. **25**(2), 441–448 (2009)
8. Babiceanu, R.F., Chen, F.F., Sturges, R.H.: Real-time holonic scheduling of material handling operations in a dynamic manufacturing environment. Robot. Comput.-Integr. Manuf. **21**(4–5), 328–337 (2005)
9. Morariu, O., Borangiu, Th., Raileanu, S., Morariu, C.: Redundancy and scalability for virtualized MES systems with programmable infrastructure. Comput. Ind. **81**, 26–35 (2016)
10. Varvara, G.: On the Team-based goal-oriented development for holonic manufacturing systems. In: Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics, Springer series Studies in Computational Intelligence, vol. 544, pp. 111–126 (2014)
11. Panescu, D., Pascal, C.: An extended contract net protocol with direct negotiation of managers. In: Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics, Vol. 544 of the Springer series Studies in Computational Intelligence, pp. 81–95 (2014)
12. Panescu, D., Pascal, C.: Holonic coordination obtained by joining the contract net protocol with constraint satisfaction. Comput. Ind. **81**, 36–46 (2016)
13. Indriago, C., Cardin, O., Rakoto, N., Castagna, P., Chacòn, E.: H2CM: a holonic architecture for flexible hybrid control systems. Comput. Ind. **77**, 15–28
14. Raileanu, S., Borangiu, Th., Ivanescu, N., Morariu, O.: Integration of mobile agents in distributed manufacturing control. In: Proceedings of the 18th International Conference on System Theory, Control and Engineering, 17–19 Oct, Sinaia, Romania (2014)
15. Kruger, K., Basson, A.H.: Implementation of an Erlang-based resource holon for a holonic manufacturing cell. Springer Stud. Comput. Intell. **594**, 49–58 (2014)
16. Bellifemine, F., Carie, G., Greenwood, D.: Developing Multi-agent Systems with JADE. Wiley (2007). ISBN: 978-0-470-05747-6
17. Morariu, C., Morariu, O., Borangiu, T., Sallez, Y.: Formalized information repre-sentation for intelligent products in service-oriented manufacturing. In: 11th IFAC Workshop on Intelligent Manufacturing Systems, Sao Paolo, Brazil, Part 1, vol. 11, pp. 318–323 (2013)
18. Borangiu, Th., Raileanu, S., Berger, T., Trentesaux, D.: Switching mode control strategy in manufacturing execution systems. Int. J. Prod. Res. **53**(7), 1950–1963 (2015). https://doi.org/10.1080/00207543.2014.935825
19. Fernández-Díaz, A., Benac-Earle, C., Fredlund, L.: Adding distribution and fault tolerance to Jason, Vol. Part. Sci. Comput. Program. **98**, 205–232 (2015)
20. Fedoruk, A., Deters, R.: Improving fault-tolerance by replicating agents. In: Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, AAMAS'02, pp. 737–744. ACM, New York (2002). ISBN: 1-58113-480-0
21. Klein, M., Rodriguez-Aguilar, J.-A., Dellarocas, C.: Using domain-independent exception handling services to enable robust open multi-agent systems: the case of agent death. Auton. Agents <AS, 179–189 (2003). http://dx.doi.org/10.1023/A:1024145408578
22. Virtual Replicated Agents Tutorial. http://jade.tilab.com/doc/tutorials/virtualAgents/index.html. Consulted in June 2017
23. Tang, Y., Sun, H., Wang, X., Liu, X.: Achieving convergent causal consistency and high availability for cloud storage. Future Gener. Comput. Syst. **74**, 20–31 (2017)
24. Shivakumar, S.K.: Ensuring high availability for your enterprise web applications. In: Architecting High Performing, Scalable and Available Enterprise Web Applications, pp. 59–99 (2015)
25. Meier, S.: Using IBM Virtualization to manage cost and efficiency. In: Redguide for Business Leaders. IBM RedBooks (2009)
26. Morariu, O., Borangiu, Th.: Resource monitoring in cloud platforms with tivoli service automation management. IFAC Proc. Vol. **45**(6), 1862–1868, (2012)
27. Dehury, C.K., Sahoo, K.: Design and implementation of a novel service management framework for IoT devices in cloud. J. Syst. Softw. **119**, 149–161 (2016)
28. Vathy-Fogarassy, Á., Hugyák, T.: Uniform data access platform for SQL and NoSQL data-base systems. Inf. Syst. **69**, 93–105 (2017)

Reconsidering the Relationship Between Cloud Computing and Cloud Manufacturing

Hélène Coullon and Jacques Noyé

Abstract History shows many relations between computer science and manufacturing control, starting with the initial idea of "digital manufacturing" in the '70s. Since then, advances in computer science have given birth to the *Cloud Computing* (CC) paradigm, where computing resources are seen as a *service* offered to various end-users. Of course, CC has been used as such to improve the IT infrastructure associated to a manufacturing control infrastructure, but its principles have also inspired a new manufacturing paradigm *Cloud Manufacturing* (CMfg) with the perspective of many benefits for both the manufacturers and their customers. However, despite the usefulness of CC for CMfg, we advocate that considering CC as a core enabling technology for CMfg, as is often put forth in the literature, is limited and should be reconsidered. This paper presents a new core-enabling vision toward CMfg, called *Cloud Anything* (CA). CA is based on the idea of abstracting low-level resources, beyond computing resources, into a set of core control building blocks providing the grounds on top of which any domain could be "cloudified".

Keywords Cloud computing · Cloud manufacturing · Resource management IaaS · MES

1 Introduction

There is already some history of using advances in computer science and information technologies to enhance manufacturing control, starting with the initial idea of "digital manufacturing" in the '70s. Since then, three breakthroughs have taken place: the interconnection of computers through the Internet, the interconnection of the software and the hardware worlds through the provision of sensors, actuators and

H. Coullon (✉) · J. Noyé
IMT Atlantique, Inria, LS2N, UBL, Nantes, France
e-mail: helene.coullon@imt-atlantique.fr

J. Noyé
e-mail: jacques.noye@imt-atlantique.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_16

embedded controllers, and finally the advent of computing as a utility through *Cloud Computing* (CC). Of course, Cloud Computing has been used as such to improve the IT infrastructure associated to a manufacturing infrastructure, but its principles have also inspired a new manufacturing paradigm *Cloud Manufacturing* (CMfg) with the perspective of many benefits for both the manufacturers and their customers in terms of efficiency, cost and flexibility. Making such a paradigm a reality remains, however, an endeavor that requires coordinated progress in many domains.

Whereas the literature shows that CMfg is often addressed through the use of CC services, thus transforming a set of manufacturing processes and functions into Manufacturing-as-a-Service (MaaS), we claim in this paper that CC, despite being useful, does not give access to the software stack at the right level (because of heavy virtualization techniques) to be well suited to the requirements of the lowest layers of CMfg, responsible for packaging the physical manufacturing resources as a service. Instead of considering CC as a core enabling technology for CMfg, we present another vision where common control building blocks, responsible for low-level resource management, could be designed by abstracting away the resource specificities. We call our vision the *Cloud Anything* (CA) model. Thus, the CA model could be used to indifferently build the lowest resource-management-centric layer of CC, i.e., the Infrastructure-as-a-Service layer (IaaS), or the equivalent lowest layer of CMfg.

The rest of the paper is organized as follows. Section 2 overviews work related to CMfg. As a foundation for the following sections, Sect. 3 compares computing and manufacturing per se. Section 4 comes back to their "Cloud" counterparts and compares them. This comparison leads to Sect. 5, which presents our Cloud Anything (CA) vision and opens a discussion about its advantages, difficulties and research challenges. Finally, Sect. 6 concludes this work and presents perspectives.

2 Related Work

As the term Cloud Manufacturing (CMfg) was coined from Cloud Computing (CC), a number of papers have studied how CMfg is related to CC with a look at future trends of CMfg.

In an early paper [19], Tao et al. present a 7-layer architecture of CMfg with a core cloud-service layer. CC is considered as a core enabling technology, to be complemented with IoT technologies together with new service layers (for instance, *Manufacturing-as-a-Service*) catering for resources that are not IT resources. These new layers are pictured as vertical, whereas the cloud services are pictured as horizontal but the precise relationship between the layers is not elucidated. This initial work is refined in [18], which distinguishes the *Internet of Things*, responsible for *service generation*, the *Internet of Services*, based on CC and responsible for *service management* (including aggregation of lower services to create the manufacturing services evoked above), and the *Internet of Users*, responsible for *service application*, i.e., on-demand use and cooperation.

In [22], Xu makes a broad comparison of CC and CMfg from low-level concerns (e.g., virtualization), up to the application layers for end-users. Xu also introduces a fundamental distinction between *smart industries*, which implies the use of CC (IoT could be brought into the picture, too) inside CMfg, in order to handle the large data sets required to take smart decisions about production lines, and CMfg per se, i.e., applying the Cloud Computing model to manufacturing processes. CMfg services are obtained by virtualizing resources made available to consumers through a Cloud platform. These resources can be tangible (they correspond to physical or basic computational resources) or intangible (they correspond to manufacturing *capabilities* [23]).

In [20], Wu et al. present another interesting survey of CMfg including a state of the art and a strategic vision of CMfg. The state of the art considers the use of CC as a *low-hanging fruit* and quotes the layered architecture of [19] but does not consider the interplay of CC and CMfg further as the research challenges focus on strategic, application-oriented concerns.

A basic issue with standard, *public* CC services is that these services are located far from the shop floor, under external management, which creates performance as well as security issues. This can be alleviated by considering using *private* CC services. This is what Morariu et al. do [12]. Shop floor devices as well as part of the Manufacturing Execution System (MES) [5] are virtualized in a private cloud whereas a public cloud is used for high-level application services. As the physical resources are seen as agents virtualized in the cloud, there is not much intelligence left in the physical layer of the architecture. Experiments show the effect of virtualization on the performance of event propagation. A broader discussion of the advantages and disadvantages of relying on public, private, community and hybrid Cloud Manufacturing solutions is available in [8]. The discussion results in the design of a Hybrid Manufacturing Cloud (HMC) infrastructure. However, performance and virtualization issues are not considered. The focus is rather on access control and interoperability issues. These issues are addressed through the use of an ontology and rule-based reasoning with an implementation hosted by a public cloud.

In [6], Kubler et al. are also concerned with interoperability issues but they include IoT in the picture. IoT is considered as another core enabler and key in *product-centric control*. Of course, interoperability requires proper generic and open IoT standards. The issue is then to seamlessly integrate IoT and CC technologies. This dichotomy between a physical layer, handled by IoT technologies, and a virtual layer, handled by CC technologies can be generalized to *Cyber Physical Systems*. In [15], Queiroz et al. consider distributed, collaborative and adaptive process supervision and control in this context. The emphasis is on data analysis, with real-time analysis for monitoring and control at the physical level, and big-data analysis for optimization, planning, and decision making at the virtual level. Interestingly, the paper talks about *smart factories* but does not mention CMfg nor virtualization. It does not address the core of CMfg as defined by Xu.

All the above papers present some interesting viewpoints on CMfg as well as some preliminary ideas on the general architecture of CMfg infrastructures. However, none of them studies the specific research challenges of resource management, which is the base of any higher-level solution (services, applications etc.). In this paper, we take the position of [22], splitting Cloud Manufacturing and Smart industries, and we present a new approach to address the lowest levels of the shop floor and MES functionalities by considering CC as a sibling domain of CMfg rather than a core enabling technology.

3 Computing and Manufacturing

Before comparing Cloud Computing and Cloud Manufacturing, let us, in a first step, forget about the cloud. The basic question is then to relate Computing Manufacturing, or, more exactly, to Manufacturing control. Here is a very simple way to look at it. Computations can be described by composing *computable functions*. A basic computation step can be informally written as $out = f(in_1, \ldots, in_k)$, taking k inputs in_i and producing an output *out* as a result of applying the function f to the inputs. This can also be seen as describing a basic manufacturing step, taking as inputs k parts and producing a new part as a result.

Of course, the nature of the inputs and outputs is quite different. They belong to the *virtual*, or digital world in the first case, they are called *data*, and the *physical* one in the second case, we will call them *parts*. Note that the virtual world does not exist per se, but is rather an abstraction of the physical world (*e.g.,* a number is a series of digits, which are themselves abstractions of electrical signals). The nature of the function f is also different. It is a *computable function* in the first case, whose execution relies on a *programmable universal machine*, *e.g.,* a *computer*. In the second case, there is no universal machine to execute a *manufacturing function*. Indeed, manufacturing requires either to rely on dedicated machines or series of such machines configured into proper assembly lines. In practice, such a process is not completely absent from computing (it occurs when compiling programs) but can easily be abstracted away. Connecting manufacturing machines and computers results in *hybrid machines*, which can take as input both parts and data and produce as output both parts and data.

All these machines can be connected at different scales through digital and production/transport networks with a huge difference: whereas data can travel at the speed of light, parts can hardly reach the speed of sound. Such a gap also exists when considering *context switching*, i.e., the possibility to interrupt the execution of a computation or a manufacturing step to schedule another one. This is very easy and fast in the virtual world, at different granularity levels, from ultra-lightweight threads to *virtual machines* (VMs), with very few constraints on when this can happen. As a result, it is very easy to share a machine to perform various computations potentially related to various applications on behalf of various users. This sharing, managed by the Operating System (OS), encompasses hardware and software resources (Central Processing Units, i.e., CPU, memory, files…). On the other hand, sharing a manufacturing machine is possible with a different latency granularity as building a part or moving a part is a much longer process.

Whereas in the computing world an OS is responsible for the automatic management of all resources (CPU, RAM, disk, network, files, etc.) within a computer and their sharing between functions, the automatic management of manufacturing resources' operations, and their sharing between manufacturing orders, has led to the design of Manufacturing Execution Systems (MESs) [5].

4 Cloud Computing and Cloud Manufacturing

The previous section has compared manufacturing control with computing, including MESs and OSes. It appears that many similarities can be found in these concepts. In this section we explore further these similarities by studying CC and CMfg and comparing them.

4.1 Cloud Computing

The National Institute of Standards and Technology (NIST) defines Cloud Computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [11].

This technical model is also closely linked to an economic model whereby users only pay for what they consume, also called the *pay-as-you-go* model. Thanks to this economic model, CC is now a widely spread model used by many companies, public or private institutes, and even by many individual persons all over the world. Actually, this economic model is responsible for an easy, cheap, permanent and seemingly unlimited access to computing and storage resources everywhere. For example, for small companies or start-ups, taking advantage of CC resources is cheaper than buying physical resources and paying IT administrators.

A cloud is composed of one or multiple large pools of distributed heterogeneous computers, called *servers*, $s_1, \ldots, s_n$, each one with its own set of resources (CPU, memory, etc.). One pool of servers is often called a data centre. To guarantee the viability of the model, the utilization of the pool of resources is optimized and can be reorganized on the fly to be able to answer new incoming user requests. If a computation f can be executed by any of the servers, its execution is specific to the target machine. For this reason, the main enabling technology of CC is *virtualization*, i.e., the ability of creating and managing *virtual machines* (VMs). This mechanism offers a way to decouple a computer (as a set of physical computing resources) from its use, thus enabling easy migration of a computation or a function f to another computer during its execution. This mechanism enables agility and an efficient way to optimize the utilization of a pool of resources. Moreover, this mechanism enables heterogeneity of computing resources, which is vital for a durable and scalable CC solution.

Thanks to CC a user can focus on its core domain without knowledge of lower-level IT concerns. Most of the time, three different abstraction levels are proposed to the user within a CC solution: (1) *Infrastructure-as-a-Service* (IaaS), where processing, network and storage resources are requested by the user under the form of virtual machines and virtual networks; (2) *Platform-as-a-Service* (PaaS), where users request complete development platforms, hence leading to the automatic provisioning of virtual machines with an initial operating system and possibly additional specific libraries, frameworks or tools proposed by the cloud provider; and (3) *Software-as-a-Service*, where users request access to a full application, which implicitly (for the user) leads to the provisioning of a complete software stack on top of a set of virtual machines.

There are different kinds of users at the different levels, from low-level developers (using IaaS), companies developing a cloud-based application by using existing development frameworks (using PaaS), or end-users running an application on their smartphone (thus using SaaS). When a higher abstraction level is defined, it is based on lower ones, the IaaS being the lowest level of CC solutions. In this paper, we focus on the IaaS level of usual CC facilities. Furthermore we claim that a set of common building blocks can be found in both a IaaS system and the lowest layers of a CMfg system by considering a resource as a broader generic notion.

A IaaS is responsible for the automatic (or semi-automatic) management and optimization of a pool of computing, storage and network resources (a data center) within a CC infrastructure. A IaaS is usually structured as a *MAPE loop*, often used in computer science [3, 10]. A MAPE loop is a perpetual loop composed of four coarse-grain steps: (1) *monitor*, to get information from the infrastructure and external inputs; (2) *analyze*, to analyze monitoring information and take decisions; (3) *plan*, to schedule decisions taken during the analysis; and (4) *execute*, to execute the plan. A MAPE loop is the basis of any autonomic system. For example, within OpenStack, the de-facto open-source solution is to address the IaaS level of the CC paradigm;...... the Telemetry project[1] represents the Monitoring step of MAPE, the Nova project[2] represents the Planning step of MAPE, etc. Thus, a IaaS system can be defined as an operating system handling CC infrastructures, i.e., pools of on-demand distributed computing, storage and network resources.

4.2 Cloud Manufacturing

Inspired by CC, which transforms a pool of IT resources into a set of on-demand services, the vision of [22] is that Cloud Manufacturing transforms a pool of manufacturing resources machines their controllers and capabilities into on-demand manufacturing services. Thus, in [22], Cloud Manufacturing is defined as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable manufacturing resources (e.g., manufacturing software tools, manufacturing

[1] https://wiki.openstack.org/wiki/Telemetry.

[2] https://wiki.openstack.org/wiki/Nova.

equipment, and manufacturing capabilities) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

CMfg can be, at least, as complex as CC, i.e., composed of multiple layers of services inherited from MESs together with an economic model. In this paper, we restrict our vision to the lowest level of manufacturing and computing systems: the resource management and its optimization. Thus, in this paper, if, for CC we focus on the IaaS layer, i.e., an operating system for distributed on-demand computing resources, for CMfg we focus on the transformation of the lowest levels of MESs to a management system for distributed on-demand manufacturing resources.

For this reason, this paper separates CMfg from CC as a start, even if both of them are related at some point. On the one hand, we consider that taking advantage of a combination of CC, Manufacturing systems and IoT techniques (getting information from small sensors), to smartly take decisions regarding production, products and related services, is not the definition of CMfg but is more related to the concept of *Smart Industry* [15]. Thus, on the other hand, we consider that CMfg refers to enabling on-demand access to manufacturing resources everywhere, which does not necessarily include the use of CC but is related to it.

As mentioned in Sect. 2, most existing CMfg initiatives suggest using CC facilities to host low- (physical layer) and high-level functionalities of MESs, as well as higher business layer functionalities [6, 12, 16]. By using CC facilities, manufacturing industries get rid of the burden of investing and maintaining IT resources locally, and also increase the quality of service regarding fault tolerance and security for cloud-hosted functionalities.

In [12, 16] private CC solutions are used to deploy low-level functions responsible for the physical layer in MESs. A private cloud offers better proximity compared to a public cloud, which leads to lower latencies and facilitates security and data privacy management (within a restricted geographical region). However, renting private CC facilities is more expensive than renting public facilities, and private resources are limited.

The authors of [16] objectively claim that by using CC the virtual machines used will always be available, because of the High Availability (HA) of CC, but this does not guarantee that the applications deployed within the VMs will automatically inherit the same availability. This illustrates that, in order to offer a complete CMfg solution, as defined in [22], with equivalent properties to those offered by CC, the same kind of development efforts have to be made, and that using CC to deploy MES functionalities in VMs does not make CMfg.

This has led us to devise a new approach to CMfg.

5 Cloud Anything

This section presents our new approach to handling resources in CMfg and, more generally, to build the lowest layers (such as a IaaS) of any Cloud system. In a first step, a centralized setting is assumed. In a second step, distribution aspects are discussed based on the emerging new paradigms for geo-distributed Clouds.

5.1 Abstraction of Resources in Low-Level Cloud Models

As explained in Sect. 4 CMfg approaches typically use CC facilities to deploy various kinds of low- and high-level functionalities of MES and business functions. A major advantage of this approach is to externalize the deployment and management of these control functions. Thus, the IT management is not handled by the manufacturing enterprise, and CC operators offer guarantees on fault tolerance, security, high availability, etc. This usual approach is represented in Fig. 1. In this figure, the 5-layer ISA-95.03 specification of MES as defined in [16] is used as an example. Most of the time, the lowest layers, the ones responsible for resource management (e.g. the PC-type workstations of machines, robots), are placed in IaaS-provided VMs. Higher layers can also be handled by CC services. The figure does not make any assumption on the type of the infrastructure (public, private…).

However, deploying applications and functions on a CC infrastructure is not sufficient to inherit all the characteristics of the CC paradigm [16]. A CC facility offers a precise type of service, i.e., an on-demand infrastructure, platform or application. When using the IaaS level, the CC operator guarantees the high availability, scalability, fault tolerance and security of rented VMs, but cannot guarantee that the same properties will apply to the application hosted on the VMs. This is why we think that implementing manufacturing functions using CC, despite being useful, does not lead by itself to CMfg. To offer manufacturing as a service, the same efforts that have been undertaken to implement CC must also be undertaken, with their own specificities, to implement CMfg.

In addition to this, we have noticed many conceptual similarities between IaaS and MES systems. In both cases a pool of distributed resources (e.g. computing resources) has to be managed according to a set of requests, computing requests in one case and production orders in the other. Hence, in both cases the concept of resource is used and common control building blocks can be found. For example, building blocks responsible for receiving requests, for computing scheduling optimizations, for receiving monitoring information from sensors, for quality management, etc., can be found in both IaaS and MES systems. For this reason, we propose another way to reach CMfg.

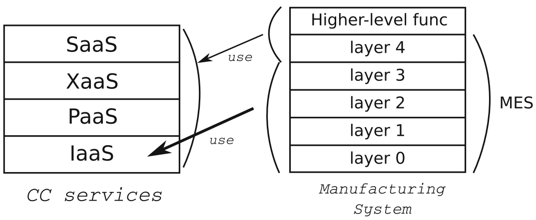


Fig. 1 Usual CMfg approach using CC to deploy all MES and business functions

Instead of deploying MES functions in a CC infrastructure using IaaS, we propose to build a common set of cloud control building blocks that, in a first step, abstract the concept of a resource, but, in a second step, can be specialized to a particular domain using meta information specific to the domain. Depending on various factors linked to the domain as well as technological constraints, this specialization could happen at modeling, implementation, or execution time. This idea is depicted in Fig. 2 where it is applied to both CC and CMfg. As in both cases an autonomic system is built, we assume that the set of common building blocks are organized around a MAPE loop. However, such an organization is quite generic. It should be refined in order to integrate some other important aspects of resource management, e.g., an economic model, required to control resource consumption and manage billing information. We call this model *Cloud Anything* (CA) as it could be applied to various domains, beyond CC and CMfg.

This model allows using CC facilities to get more computing resources if needed, as implied in the figure by the thin arrow. Still, the lowest layers of MESs do not need to be deployed using IaaS when high performance is required. IaaS is a very high-level on-demand computation service, resulting in high overheads (due to data transfers through the network, distant data centres, and complex software stacks for VM management). Our proposal offers a generic way to build a Cloud Anything infrastructure, decoupled from the underlying resources. Moreover, it offers the possibility to handle both CC and CMfg within a single operator with minor overheads.

On the other hand, what happens if resources embed computing resources, e.g., a computer unit responsible for hosting control or a product with embedded intelligence? Does this computing resource, used within the IaaS-like implementation layer, require specific administration without any support from the infrastructure? Such a situation already happens in CC where deploying a IaaS layer requires computing resources. Many solutions are appearing to address such a situation. These solutions consist in the automation of the deployment of IaaS components onto distributed

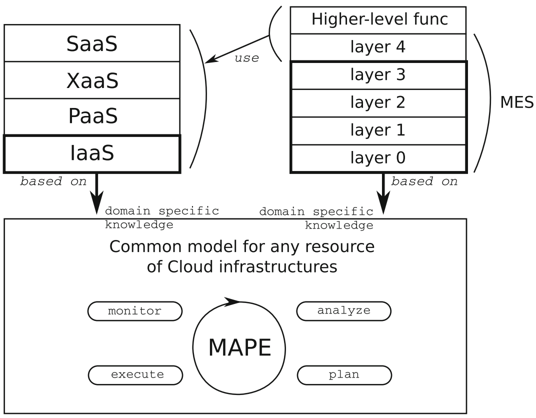


Fig. 2 New approach to CMfg, the Cloud Anything model

computers [2] (Juju,[3] Kubernetes[4] [17], TripleO,[5] etc.). The weird and touchy part of deployment automation is that such a system is close to a IaaS system itself (as a IaaS is responsible for the deployment and management of virtual machines), but the system can be much lighter than a IaaS as all Cloud-related parts of a IaaS do not need to be handled. Moreover, such solutions can work without virtual machines (namely bare-metal resources), or can use lighter virtualization techniques such as containers[6] [4] or uni-kernels [21].

To get back to our proposal depicted in Fig. 2, many research challenges, in addition to more technical issues, arise from abstracting resources from a set of resource management building blocks, thus from externalizing any specific knowledge from the type of resource. However, such work could be a great advance to easily move toward various kinds of Cloud infrastructures.

5.2 Advances in Geo-Distributed Clouds

Fog and Edge Computing [1, 9, 14] are two rather new CC paradigms. They extend the centralized CC concept with a massively geo-distributed set of micro Clouds deployed on strategic points inside the highly-connected Internet network and into small intelligent objects. Hence, CC is extended with Fog (core network micro data centres) and Edge (smart objects used as small CC resources) facilities, which are closer to data sources and end-users. Both paradigms can be classified as massively geo-distributed utility computing.

Many advantages can be gained from the Fog and the Edge Computing paradigms. First, one outcome and drawback of centralized CC is the huge amount of energy consumed by the infrastructure itself, but also by cooling facilities [13]. Thus, by limiting the size of centralized CC and by adding many small Clouds into the highly connected infrastructure, the energy consumption can be more easily handled. Second, like private CC, these paradigms can improve latency and bandwidth issues, compared to centralized distant CC, and can help controlling security or data privacy.

However, research challenges also arise from the design of Fog and Edge infrastructures, particularly when dealing with massively geo-distributed control building blocks of IaaSes for very heterogeneous resources [2, 7].

Interestingly, CMfg also needs to be designed as a geo-distributed Cloud managing very heterogeneous kinds of resources and gathering multiple sites of manufacturing to aggregate enough resources for elasticity, high availability, etc. The work on distributed MESs and their design as multi-agent holonic systems (see, for instance, [12, 15, 16]) is very relevant in this context, with some shared research challenges

[3] https://jujucharms.com/.

[4] http://kubernetes.io/.

[5] https://wiki.openstack.org/wiki/TripleO.

[6] https://www.docker.com/.

appearing when designing a massively distributed IaaS and a cloud-based holonic MES. In other words, considering our proposal, the set of generic building blocks could also be designed such that each block can be distributed, thus making it possible to manage geo-distributed complex Cloud Anything infrastructures. Moreover, at higher levels, including Smart industries, where CC is used to conduct heavy computations, upgrading to Fog and Edge CC seems to be a very good choice to get lower latencies and stronger security properties. There is a potential for holonic systems to inspire IaaSes under design for Fog and Edge computing as well as the other way around. This, in addition to our proposal, represents a great opportunity for collaborative work.

6 Conclusion

This paper has explored the relationship between computing and manufacturing control and between Cloud Computing and Cloud Manufacturing, from a Computer Science perspective, with a focus on the lower service-level of resource management.

This has suggested considering the design of a generic resource management layer compatible with a wide range of resources and generalizing the IaaS layer of Cloud Computing beyond computing resources. This is a first step toward a generalization of the concept of Cloud, enabling "Cloudification" of various domains including Manufacturing. This also makes it possible to handle both Cloud Computing and Cloud Manufacturing within a single operator, with minor overheads compared to usual approaches. The design of such a generic layer is challenging but could benefit from existing results and on-going work on both Cloud Computing and Cloud Manufacturing with synergistic effects.

The next step would be to identify the common control building blocks of resource management for Clouds (Computing and Manufacturing) and explore how to implement our proposal on an appropriate subset in order to produce a proof of concept, with a particular attention paid to a distributed (or holonic) design.

References

1. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, MCC '12, pp. 13–16. ACM Press (2012)
2. Coullon, H., Pérez, C., Pertin, D.: Production deployment tools for IaaSes: an overall model and survey. In: 2017 IEEE International Conference on Future Internet of Things and Cloud (FiCloud), Prague, Czech Republic (2017)
3. Huebscher, M.C., McCann, J.A.: A survey of autonomic computing—degrees, models, and applications. ACM Comput. Surv. **40**(3), 7:1–7:28 (2008)
4. Joy, A.M.: Performance comparison between linux containers and virtual machines. In: 2015 International Conference on Advances in Computer Engineering and Applications, pp. 342–346 (2015)

5. Kletti, J.: Manufacturing Execution System—MES, 1st edn. Springer (2010)
6. Kubler, S., Holmström, J., Främling, K., Turkama, P.: Technological theory of cloud manufacturing. In: Borangiu, T., Trentesaux, D., Thomas, A., McFarlane, D. (eds.) Service Orientation in Holonic and Multi-Agent Manufacturing, Studies in Computational Intelligence, vol. 640, pp. 267–276. Springer (2016)
7. Lebre, A., Pastor, J., Simonet, A., Desprez, F.: Revising OpenStack to operate fog/edge computing infrastructures. In: IEEE International Conference on Cloud Engineering, Vancouver, Canada (2017)
8. Lu, Y., Xu, X., Xu, J.: Development of a hybrid manufacturing cloud. J. Manuf. Syst. **33**(4), 551–566 (2014)
9. Mahmud, R., Buyya, R.: Fog computing: A taxonomy, survey and future directions. CoRR (2016). arXiv:abs/1611.05539
10. Maurer, M., Breskovic, I., Emeakaroha, V.C., Brandic, I.: Revealing the MAPE loop for the autonomic management of cloud infrastructures. In: 2011 IEEE Symposium on Computers and Communications (ISCC), pp. 147–152 (2011)
11. Mell, P.M., Grance, T.: The NIST definition of cloud computing. Special Publication 800-145. National Institute of Standards & Technology, Gaithersburg, MD, United States (2011)
12. Morariu, O., Borangiu, T., Morariu, C.: From service oriented to cloud powered manufacturing systems. In: Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, UCC '13, pp. 83–90 (2013)
13. Orgerie, A.C., de Assunção, M.D., Lefèvre, L.: A survey on techniques for improving the energy efficiency of large-scale distributed systems. ACM Comput. Surv. **46**(4), 47:1–47:31 (2014)
14. Perera, C., Qin, Y., Estrella, J.C., Reiff-Marganiec, S., Vasilakos, A.V.: Fog computing for sustainable smart cities: a survey. ACM Comput. Surv. **50**(3), 32:1–32:43 (2017)
15. Queiroz, J., Leitão, P., Oliveira, E.: Industrial Cyber Physical Systems Supported by Distributed Advanced Data Analytics, pp. 47–59, vol. 694 of Borangiu et al. [2] (2017)
16. Răileanu, S., Anton, F., Borangiu, T.: High Availability Cloud Manufacturing System Integrating Distributed MES Agents, pp. 11–23, vol. 694 of Borangiu et al. [2] (2017)
17. Rensin, D.K.: Kubernetes-Scheduling the Future at Cloud Scale. O'Reilly (2015)
18. Tao, F., Cheng, Y., Xu, L.D., Zhang, L., Li, B.H.: CCIoT-CMfg: Cloud computing and internet of things-based cloud manufacturing service system. IEEE Trans. Ind. Inf. **10**(2), 1435–1442 (2014)
19. Tao, F., Zhang, L., Venkatesh, V.C., Luo, Y., Cheng, Y.: Cloud manufacturing: a computing and service-oriented manufacturing model. Proc. Inst. Mech. Eng. Part B J. Eng. Manuf. **225**(10), 1969–1976 (2011)
20. Wu, D., Greer, M.J., Rosen, D.W., Schaefer, D.: Cloud manufacturing: strategic vision and state-of-the-art. J. Manuf. Syst. **32**(4), 564–579 (2013)
21. Xavier, B., Ferreto, T., Jersak, L.: Time provisioning evaluation of KVM, Docker and unikernels in a cloud platform. In: 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 277–280 (2016)
22. Xu, X.: From cloud computing to cloud manufacturing. Robot. Comput.-Integr. Manuf. **28**(1), 75–86 (2012)
23. Zhang, L., Luo, Y., Tao, F., Li, B.H., Ren, L., Zhang, X., Guo, H., Cheng, Y., Hu, A., Liu, Y.: Cloud manufacturing: a new manufacturing paradigm. Enterp. Inf. Syst. **8**(2), 167–187 (2014)

On Increasing Adaptability of Holonic Systems

Carlos Pascal and Doru Panescu

Abstract This paper discusses about increasing the adaptability of holonic manufacturing systems. This could be obtained by a coordination mechanism that is able to solve rescheduling problems caused by unexpected events occurring at the resource level. Moreover, a learning process was considered in order to increase the performance of coordination protocol. The proposed approach is verified with a case study regarding a scheduling problem. Simulation experiments were done by the means of a Coloured Petri net model; some advantages of the proposed model with respect to the flexibility of its use are underlined, too.

Keywords Holonic manufacturing systems · Coloured petri nets
Distributed constraint satisfaction problems · Adaptability

1 Introduction

For present manufacturing systems one important, requested feature is adaptability to unpredicted events [1, 2]. About this, nowadays standards (for example, Industry 4.0) impose a transfer of the decisional and intelligent parts towards the low levels of the control architecture [3, 4]. Moreover, the new intelligent components have to be properly connected and controlled trough appropriate coordination protocols, so that the desired adaptability is obtained. Such a paradigm comes close to Holonic Manufacturing Systems (HMSs), which are developed by means of multiagent systems [5–7]. This present paper is included in this framework, namely it treats the

C. Pascal (✉) · D. Panescu
Department of Automatic Control and Applied Informatics,
"Gheorghe Asachi" Technical University of Iasi, D. Mangeron 27,
700050 Iasi, Romania
e-mail: cpascal@ac.tuiasi.ro

D. Panescu
e-mail: dorup@ac.tuiasi.ro

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_17

case of HMSs that are able to comply with changes appeared in the manufacturing environment during the production process, in order to obtain a greater adaptability.

This work continues our previous research on coordination in HMSs; our focus has been on finding an efficient method to solve planning and scheduling for a manufacturing system that uses holonic control architecture. Our proposal is about coupling two techniques used with multiagent systems: the Contract Net Protocol (CNP) and Distributed Constraint Satisfaction Problem (DisCSP) [5, 8–10]. This paper introduces a new perspective for our proposed holonic control scheme, namely the case when the manufacturing system must face an unexpected change appeared during production with respect to manufacturing resources (for example, machine tools, robots, etc.). We tackled such a situation, considering the following premises:

- Holons that are in higher positions (order and product holons [7, 11]) in the holarchy being affected by the unexpected event use a priori known plans to be applied and the change (adaptation) to be made is about a new scheduling of resource holons, so that the production goal to be achieved in the new manufacturing context.
- Resource holons are able to detect the change of their state that prohibits them to have the normal behaviour; moreover, the holonic communication system allows resource holons to inform the order/product holons about their new state.
- The HMS must treat both the case when the unexpected event related to resource holons appears before planning/scheduling is started, and the more difficult case, when this happens after that instant, meaning during execution.
- We studied only cases when the HMS has at least one solution for the problem to be solved after the appearance of an unexpected event; even so, the DisCSP mechanism can handle problems with no solution and provide the corresponding answer.
- We were interested only in finding a mechanism that can provide a solution for a scheduling problem and not in analyzing the optimization issues.

The paper brings certain new research issues in comparison with our previous work; these are:

- We studied and compared three DisCSP algorithms, i.e. Synchronous and Asynchronous Backtracking (SBT and ABT), and Weak Commitment Search (WCS) (see [12, 13]) with respect with their behaviour when applied on a classical toy problem (n-queens). Now, the DisCSP method is used for manufacturing problems by involving the holonic approach, the focus being on a learning mechanism coupled with DisCSP.
- While in our previous research ABT was involved [5, 9, 14], now we use WCS. The differences are about the priorities of agents (these are dynamically changed in WCS) and the way the variables' assignment is done with the minimization of violated constraints (the min-conflict mechanism is used [13]).
- In our earlier work we studied how DisCSP can be implicated in CNP, both with respect to a negotiation between contractors [8] and between managers [5]; the new issue introduced by this paper is about the triggering condition for

re-negotiation that is now a change appeared at the level of contractors (e.g., a resource changes its commitment, becoming unavailable for certain time intervals); in [5] the analysis was about a change appeared at the level of managers, for example as caused by the appearance of new manufacturing commands.

As about related work on the subjects of this paper, according to our knowledge the combination between DisCSP and CNP used to solve manufacturing problems was not studied in literature. In [15], the possibility of using CSP for industrial applications is invoked, but the method is implied in its centralized version. A state of the art regarding the problems related with scheduling issues for manufacturing can be found in [2]; here, one is informed on the main challenges for solving industrial scheduling problems which become difficult to handle when the system must deal with unexpected events. Our paper can be considered as a further step towards solving this issue. A review of literature on rescheduling was included in [5].

This paper is organized as follows. In the next two sections the coordination mechanism and the Petri net model used in simulations are introduced. Then, the coordination method is tested on a scheduling case study, with the focus on determining its utility for rescheduling and on analyzing the improvement that can be obtained with a learning process included in the DisCSP algorithm. Some conclusions and future work end the paper.

2 The Coordination Mechanism

As mentioned in the previous section, the considered coordination mechanism is based on a combination between the CNP [16] and DisCSP [13]. This is utilized to solve manufacturing planning and/or scheduling. The used notations are as follows (see Fig. 1). M1 ÷ Mn are managers, C1 ÷ Cm are contractors [16, 17]. Each manager (representing an order or product holon) has to assign contracts to one or more resources, so that its goal is achieved. As shown in Fig. 1, in the coordination process there are three steps: goal announcing/bidding, DisCSP based negotiation and contracting. Briefly speaking, managers apply the bidding phase in order to find contractors for their goals, and then they take into account the received bids and negotiate so that all constraints regarding resources (contractors) are satisfied. To be more specific, for the scheduling problems (as considered in this research), contractors' bids contain information on actions they can solve, the time interval when they are available and the duration of actions. The negotiation is conducted according to DisCSP and has as result the decision about the way to award contracts. In our case, a contract specifies the starting time for a resource's action, this being the variable to be assigned through DisCSP. The negotiation takes into account two types of restrictions: ordering constraints (these are known by managers according to the plans they apply) and overlapping constraints (a resource can do a single action at any instant), which are known according to the received bids. After the DisCSP based negotiation, the contracts are awarded based on the CNP mechanism (see Fig. 1). Thus, execution can begin, resources starting their activities according to the received contracts.

This research treats a new case. Namely, during execution, it can happen that some resources become unavailable (for example, due to the appearance of a malfunction or maintenance operation). In Fig. 1, the resource represented by Cj is the one that sends a message for contract cancelling, due to a new appeared constraint. In such a case, the HMS should restart the coordination process. Managers take into account the received information as new constraints and begin a new negotiation. To facilitate this new coordination phase, the involved resource should announce all managers that can use it about the time interval when it is not available. The effect will be that some contracts will be cancelled while others are still kept. Through the DisCSP procedure a new solution should be found, with new awarded contracts (the right part of Fig. 1).

An improvement that was taken into account in this paper is about a learning process that is included in the DisCSP phase (that is why in Fig. 1 it appears DisCSP with learning negotiation). This can be explained as follows. During the search of a solution for a problem instance, DisCSP mechanism discovers new constraints: the so called *nogoods* [13]. In our approach, learning means that these

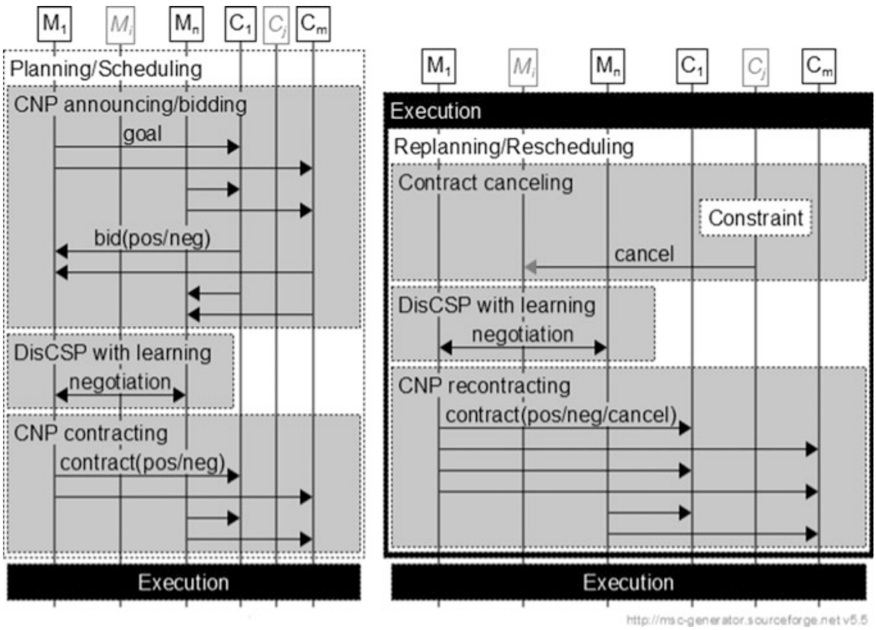


Fig. 1 Coordination protocol with re-negotiation triggered by contractors

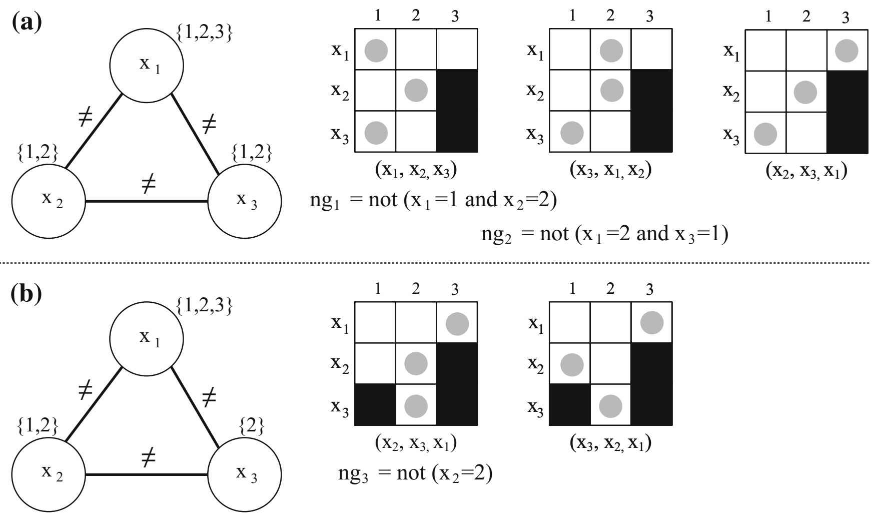


Fig. 2 An illustrative example for DisCSP with learning

found nogoods are kept and used when another instance of a problem is treated; this new instance appears when one or more constraints are added to the previous ones. As considered above, such a case appears in the coordination protocol when a resource becomes unavailable, and the knowledge from the previously carried out negotiations is used in the new coordination step.

An illustrative example for DisCSP with learning is presented in Fig. 2. The first instance of the problem is displayed in Fig. 2a; it regards three agents (x_1, x_2, x_3) with their specified domains and the constraint is that their values must be different. Such a problem can appear in a scheduling scenario, when the values are about the starting times for three actions. When the search process is conducted according to WCS [13], as shown in Fig. 2a, in the first phase the order of agents is x_1, x_2, and x_3 and the chosen values are, for example, $x_1 = 1$, $x_2 = 2$; according to these values, the agent x_3 discovers the nogood $ng_1 = \text{not}(x_1 = 1 \text{ and } x_2 = 2)$. Agent x_3 chooses the value $x_3 = 1$ and it increases its priority in order to be the first one (see the second phase). According to agents' priorities, x_1 changes its value (it becomes $x_1 = 2$) and afterwards x_2 discovers a new nogood ng_2 and increases its priority. In the last phase, x_1 assigns a new value ($x_1 = 3$) and the searching process ends with a solution. In Fig. 2b a new instance of the problem is presented that was obtained by adding a new constraint, namely the domain for x_3 is restricted to one value. Data from the previous search are kept (learned), namely agents' priorities and discovered nogoods. Agent x_3 finds a new nogood ng_3. Agent x_1 chooses the value $x_1 = 3$ by taking into account the learned nogood ng_1. Thus, the searching process is pruned, and in the second phase the solution is already found (see Fig. 2b). This example illustrates how learning was included in the DisCSP based negotiation, as will be further discussed in Sect. 4.

3 A Brief Description of the CPN Model

In the same way as in our previous research, a Coloured Petri net (CPN) model was used in order to analyse the coordination mechanism [12, 18]. Figure 3 represents the model for the DisCSP based negotiation phase, which is detailed in [12]. This is devoted to three types of DisCSP algorithms, namely ABT, SBT and WCS. The common notation for Petri nets is used, with rectangles representing transitions and ellipses for places [18]. Transitions model events and places are parts of agents' knowledge bases. This model represents all agents, each agent having its corresponding token in the places of the CPN (in Fig. 3, the number of agents is eight, which corresponds with the number of actions in the scheduling case study that will be presented in the next section). Exceptions are places dom1 and dom2 that are used in variables' assignment according to DisCSP. A transition becomes active and can be fired if all its input places contain the information that validates the guard function being attached to that transition. Each transition has an action function that carries out the event; it correspondingly changes the agents' knowledge bases. In comparison with the previously presented model (see [12]), now a new place (domain) is introduced.

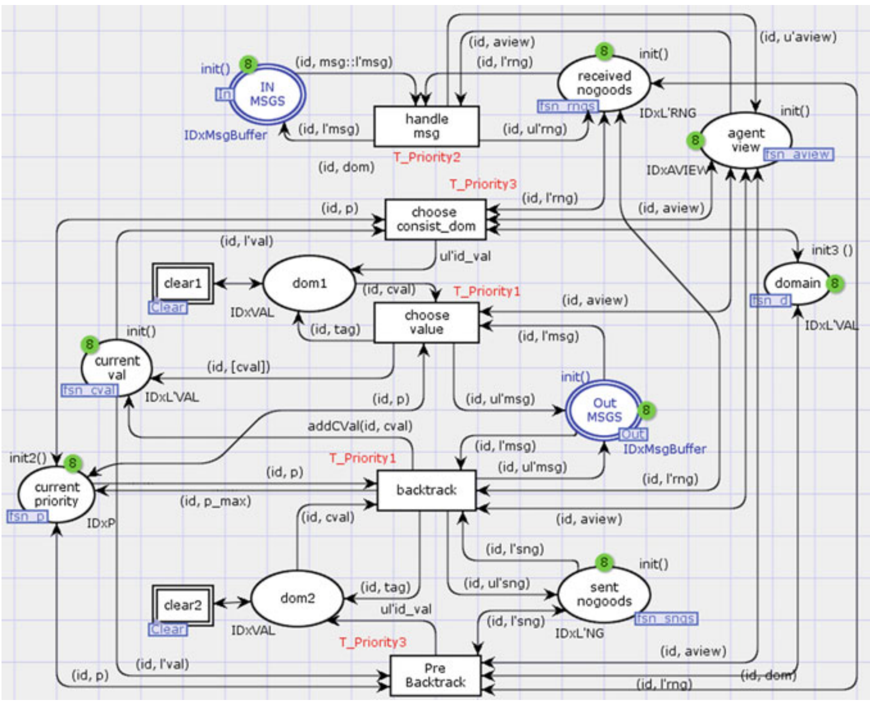


Fig. 3 Coloured Petri net model for DisCSP based negotiation

This allowed us (as it will be explained below) to make a larger range of experiments by modifying the variables' domains of the CSP method in accordance with new added constraints.

As explained in the previous section, we wanted to materialize a learning process during the DisCSP phase. Taking into account this purpose, the model was enhanced as one can see in Fig. 4. A new transition was added (named next episode) that will be activated when a negotiation process is ended. Its result regards the introduction of new restrictions before starting a new episode. Thus, our CPN could model the behaviour of a new coordination process started with new constraints as imposed by resources. One can see that transition next episode does not modify the places received, sent nogoods, and current priority, which means that all information from previous episodes is kept and the learning process takes place. The place no_of_episodes is used to indicate the number of episodes in a series of simulation experiments. This place contains only one token, which is a number used when testing the moment for stopping the simulations.

In Figs. 3 and 4, places with the same names are identical, and they were represented in a distinct way to become easier to understand; this is possible by using the fusion mechanism provided by the CPN Tool [18]. The priorities of transitions are marked in red. Transition next episode has the lowest priority because it must be fired only in the end of a searching process. The CPN model was developed by the means of CPN Tool and it is available at [19].

As specific issues regarding the described model, in order to reduce the number of messages exchanged between agents, these are grouped. Namely, when an agent sends the information about the chosen value, this is accompanied by the new discovered constraints (nogoods) [12]. Furthermore, because WCS is the used

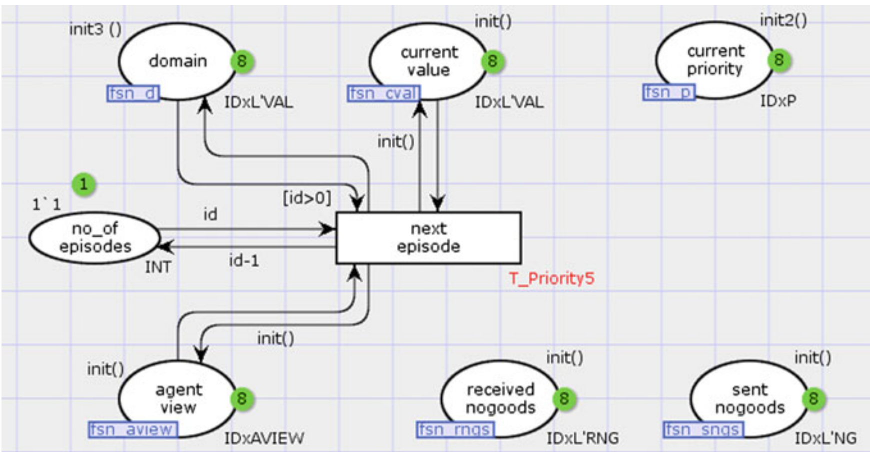


Fig. 4 Extended model allowing the study of a learning process

algorithm, a value is selected by an agent in order to minimize the number of conflicts with other agents, in accordance with agents' priorities that are dynamically changed. In Appendix 1 one can see the model declarations for the scheduling problem that is treated in the next section.

4 A Scheduling Case Study

A scheduling problem was considered as case study, namely the one described in [20] and represented in Fig. 5. It is about manufacturing of three products; these are handled by managers $M_1 \div M_3$. The plans for products are known and they contain eight actions ($a_1 \div a_8$ in Fig. 5). It is to remark that agents appear in our approach in two perspectives. On one hand, there are holonic agents (that can have the role of managers and contractors), and on the other hand there are action agents that are used in the DisCSP phase. Namely, there is such an agent for each action, which simplifies the application of CSP mechanism, because each agent handles a single variable. Contractors are resources (e.g. machine tools), being marked with C_i. Ordering constraints are represented by arrows, while overlapping restrictions (actions carried out by the same resource) are displayed as arcs. All actions need one time unit, excepting for action a_1 and a_7 that are carried out in two time units; with these specifications, the scenario has eighteen solutions for schedules with five time units [9]. One such solution is presented in the bottom part of Fig. 5.

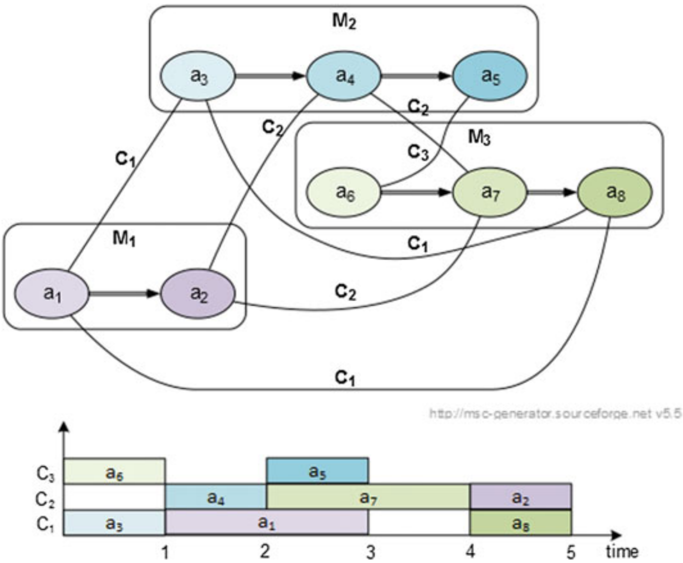


Fig. 5 A case study of scheduling problem

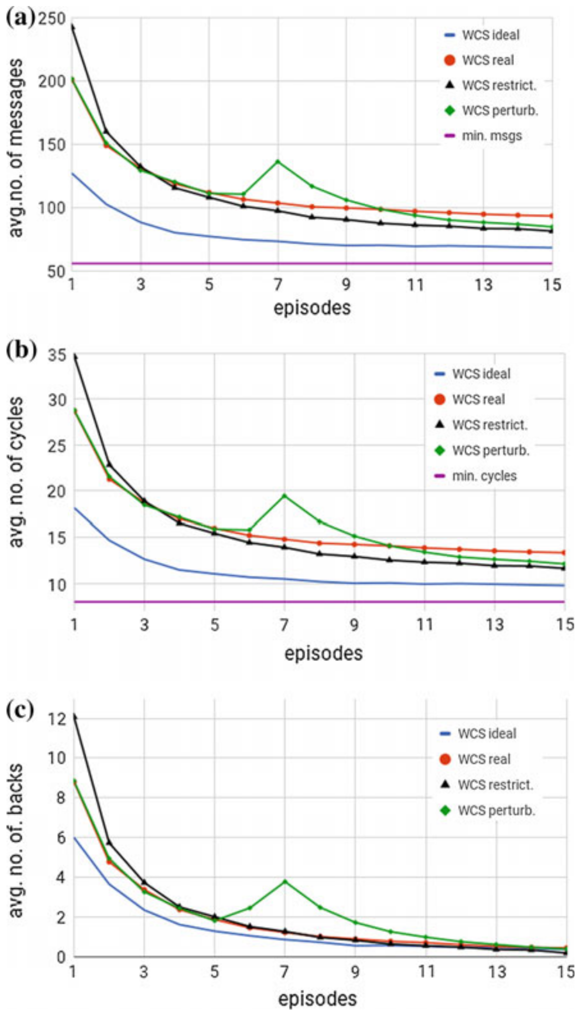


Fig. 6 Average results for series of 1000 simulation experiments; **a** number of messages **b** number of cycles **c** number of nogoods

As already mentioned, from the class of DisCSP algorithms, WCS was used. The performance parameters being analysed were the communication load (number of messages) and computational load (number of cycles and number of back-tracking events [12]).

Figure 6a, b and c represent the results for series of 1000 simulations; the values of the displayed parameters are the average ones. First, the behaviour with respect to the learning process was studied. In each simulation 15 episodes were involved.

The ideal case, when all agents choose the right value from the beginning is displayed in Fig. 6 as the *min* value. The curve labelled *WCS ideal* refers the case when in the DisCSP algorithm all agents take decisions based on updated, right information (this supposes that first all messages are read and only then a decision is taken). This case could be modelled with the developed CPN by assigning the highest priority to the transition *handle msg* (see Fig. 3). Opposed to this, the case happening with real multiagent systems, namely when an agent decides using both updated and obsolete information on the states of other agents is considered, too. This case is represented in Fig. 6 by the curve labelled *WCS real*; this situation was obtained when the priority of transition *handle msg* is lower than the priorities of transitions *choose consist_dom* and *pre backtrack* (the details on these transitions can be found in [12]).

Secondly, the effect of a perturbation during execution was studied. Thus, cases labelled with *WCS retrict* and *WCS perturb* regard situations when a new constraint is added (either from the first episode—*WCS restrict*, or beginning with the 6th episode—*WCS perturb*). One can see that in all cases the learning process has a positive effect: both the communication and computational load are decreasing with the number of episodes. As expected, when a new restriction appears during execution (i.e., in our experiments resource C_1 becomes unavailable in the time interval [4, 5]—Fig. 5), an instant increase of the communication/computational load for the multiagent system is determined, but this effect is diminished in the next episodes through learning.

Another experiment was conducted for the case when the unexpected event appears in the second episode, with the results presented in Table 1. As one can see, the learning process has a positive effect again, namely for our system the perturbation does not produce an increasing of the communication/computational loads. In Table 1, the position labelled Episode 1 refers the first negotiation done in the HMS. The position Episode 2 is about the case when resource C_1 becomes unavailable for the final interval of scheduling and the knowledge acquired during the first episode is used.

The position New Episode regards the case when the resource C1 changes its state, but the learning process is not involved. According to the data of Table 1, the learning process made the search after perturbation to be similar with the case of the normal context (the average number of backtracking events, #backs in Table 1,

Table 1 A comparison for the DisCSP phase with and without learning

	Episode 1		Episode 2		New episode	
	min/max	avg.	min/max	avg.	min/max	avg.
#msgs	56/455	200.998	70/448	198.275	84/497	237.986
#cycles	8/65	28.714	17/64	28.325	12/71	33.998
#backs	0/29	8.862	0/30	8.857	0/31	11.716

is eloquent about this, being almost the same for the cases Episode 1 and Episode 2), while in the case when learning is not implied, an increased search process appears (see Table 1).

It is to remark that learning may have as a negative effect a growth of memory consumption, as needed to keep the discovered nogoods. In our approach, this consequence is diminished, because we remove from agents' knowledge bases the weak nogoods (those nogoods that define restrictions already included in stronger nogoods—see [12]).

5 Conclusion and Future Work

This paper had as goal to study a holonic coordination mechanism able to deal with rescheduling problems, determined by unexpected events in the manufacturing environment appearing at the level of resources. The new proposed coordination mechanism was able to deal with such problems, as proved with a case study taken from literature. Besides this, we wanted to see how a learning mechanism can improve the search process for finding the scheduling solution.

Even if CNP is simple and quite often used as coordination protocol, when applied in HMSs it conducts to difficulties about solving conflicts that appear both at the managers and contractors (resources) levels. In this paper, the case of a negotiation between managers is treated. This is started by information received from resources, and then managers adapt their schedules in order to comply with the new appeared constraints. The learning mechanism that we included in the DisCSP-based coordination proved to reduce the computational and communication loads when scheduling is to be re-done.

Though the literature contains many tools allowing the simulation of distributed systems, our CPN model brings new features and advantages. It is an instrument that can easily deal with different manufacturing scenarios for distributed control schemes. It can easily accommodate a different number of holons/agents, and new constraints can be added in a facile way. It also has the advantage of modeling distinct behaviours which are changed by assigning appropriate priorities to certain events. The limits of the model are related to the complexity of manufacturing problem. If the number of agents increases (the number of states in the search process will be also high) the simulation time grows exponentially. Even so, results for simple problems can be scaled for more complex ones.

As future work, it is to remark that a present trend in Artificial Intelligence is about using various learning mechanisms. Thus, after the first step made with this research, we intend to enhance the range of applying learning procedures in order to increase the adaptability of HMSs. This can be supported by the new tendencies regarding the acquisition of data from different stages of industrial processes.

Appendix 1

Model declarations for scheduling problem

```
                                   (* some declarations of types *)
colset ID = INT;
colset L'ID = list ID;
colset VAL_JobShop = product ID*INT*INT; (* resource id, start time, duration *)
colset VAL = VAL_JobShop;
colset L'VAL = list VAL;
colset P = INT;
colset IDxVALxP = product ID*VAL*P;
colset L'IDxVALxP = list IDxVALxP;
colset AVIEW = L'IDxVALxP;
                                   (* some declarations of variables *)
var id:ID;
var cval:VAL;
var aview:AVIEW;
                                   (* constraints *)
fun isConsistJobShop
      (xi:ID, (ri,di,si):VAL_JobShop, _) (xj:ID,(rj,dj,sj):VAL_JobShop, _) =
      ordering(xi,(ri,di,si),xj,(rj,dj,sj)) andalso overlap((ri,di,si), (rj,dj,sj));
val isValConsist = isConsistJobShop;
                                   (* initialization *)
val agents:L'ID = [1,2,3,4,5,6,7,8];
val ordering_constraints =[ (1,2),(3,4),(4,5),(6,7),(7,8)];
val D1:L'VAL = [(1,2,0),(1,2,1),(1,2,2),(1,2,3),(1,2,4)];
val D2:L'VAL = [(2,1,0),(2,1,1),(2,1,2),(2,1,3),(2,1,4)]
val D3:L'VAL = [(1,1,0),(1,1,1),(1,1,2),(1,1,3),(1,1,4)];
val D4:L'VAL = [(2,1,0),(2,1,1),(2,1,2),(2,1,3),(2,1,4)];
val D5:L'VAL = [(3,1,0),(3,1,1),(3,1,2),(3,1,3),(3,1,4)];
val D6:L'VAL = [(3,1,0), (3,1,1), (3,1,2), (3,1,3), (3,1,4)];
val D7:L'VAL = [(2,2,0), (2,2,1), (2,2,2), (2,2,3),(2,2,4)];
val D8:L'VAL = [(1,1,0), (1,1,1), (1,1,2), (1,1,3), (1,1,4)];
val no_of_episodes:INT = 15;
```

References

1. Keddis, N., Burdalo, J., Kainz, G., Zoitl, A.: Increasing the adaptability of manufacturing systems by using data-centric communication. In: Emerging Technology and Factory Automation (ETFA), pp. 1–8 (2014)
2. Cardin, O., Trentesaux, D., Thomas, A., Castagna, P., Berger, T., El-Haouzi, H.B.: Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges. J. Intell. Manuf. 1–15 (2015)

3. Wang, S., Wan, J., Zhang, D., Li, D., Zhang, C.: Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. Comput. Netw. **101**, 158–168 (2016)
4. Lee, J., Bagheri, B., Kao, H.A.: A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manuf. Lett. **3**, 18–23 (2015)
5. Pascal, C., Pănescu, D.: On rescheduling in Holonic manufacturing systems. In: Borangiu T., Trentesaux D., Thomas A., Leitão P., Oliveira J. (eds.) Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in Computational Intelligence, Springer, vol. 694, pp. 201–213 (2017)
6. Borangiu, T., Raileanu, S., Trentesaux, D., Berger, T., Iacob, I.: Distributed manufacturing control with extended CNP interaction of intelligent products. J. Intell. Manuf. **25**(5), 1065–1075 (2014)
7. Pănescu, D., Pascal, C.: On a holonic adaptive plan-based architecture: planning scheme and holons' life periods. Int. J. Adv. Manuf. Technol. **63**(5), 753–769 (2012)
8. Pănescu, D., Pascal, C.: Holonic coordination obtained by joining the contract net protocol with constraint satisfaction. Comput. Ind. **81**, 36–46 (2016)
9. Pascal, C., Pănescu, D.: On applying DisCSP for scheduling in holonic systems. In: 20th International Conferenvce on System Theory, Control, and Computing (ICSTCC), pp. 423–428 (2016)
10. Panescu, D., Pascal, C.: A constraint satisfaction approach for planning of multi-robot systems. In: Proceeding of 18th International Conference on System Theory, Control, and Computing (ICSTCC), pp. 157–162 (2014)
11. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.: Reference architecture for holonic manufacturing systems: PROSA. Comput. Ind. **37**(3), 255–274 (1998)
12. Pascal, C., Pănescu, D.: A coloured Petri net model for DisCSP algorithms. Concurr. Comput. Pract. Exp. (2017). https://doi.org/10.1002/cpe.4179
13. Yokoo, M., Durfee, E., Ishida, T., Kuwabara, K.: The distributed constraint satisfaction problem: formalization and algorithms. IEEE Trans. Knowl. Data Eng. **10**(5), 673–685 (1998)
14. Pascal, C., Pănescu, D.: On improving efficiency of DisCSP Methods. In: 21st International Conference on System Theory, Control, and Computing. Sinaia, Romania (2017). https://doi.org/10.1109/ICSTCC.2017.8107015
15. Ghédira, K.: Constraint satisfaction problems: CSP formalisms and techniques. Wiley (2013)
16. Smith, R.G.: The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Trans. Comput. **12**, 1104–1113 (1980)
17. Pănescu, D., Pascal, C.: An extended contract net protocol with direct negotiation of managers. In: Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics, pp. 81–95. Springer (2014)
18. Jensen, K., Kristensen, L.M.: Coloured Petri Nets: modeling and Validation of Concurrent Systems. Springer, New York (2009)
19. DisCSP Models. http://github.com/carlospascal/DisCSP-CPN-Models (2017)
20. Bongaerts, L., Monostori, L., McFarlane, D., Kádár, B.: Hierarchy in distributed shop floor control. Comput. Ind. **43**(2), 123–137 (2000)

Software-Defined Networking-Based Models for Secure Interoperability of Manufacturing Operations

Radu F. Babiceanu and Remzi Seker

Abstract In recent years, cloud manufacturing, together with cyber-physical systems and shop-floor virtualization, acts as a disruptor from traditional manufacturing operations, and allows moving from production-oriented manufacturing to customer- and service-oriented manufacturing networks. As cloud manufacturing operations increase in volume and complexity, there is a need to better define the actual network infrastructure. While exiting IP networks, with adequate security measures in place, are appropriate for cloud applications, enhanced approaches such as software-defined networks are already in use for several domains. Given the sensitivity and timely requirements of manufacturing packet transmissions, software-defined networks are a promising solution for manufacturing operations, as well. Moreover, as cloud manufacturing operations expand to different actors, interoperability factor also becomes a critical requirement. Legacy and proprietary systems need to communicate with each other in a secure and timely fashion, which requires common standards and practices in place. This paper proposes a manufacturing interoperability framework built on the software-defined networking principles that can be easily programmed from the packet transmission point of view. Related to manufacturing packet route programming and its advantages compared to IP routing algorithms, the security aspects of the software-defined networks are also emphasized. Legacy control systems were built for limited frequent software changes or updates. However, networking legacy systems with the new cyber-physical systems for cloud manufacturing applications requires frequent system and security updates. Software-defined networking enables the needed system updates and thus enhances interoperability.

Keywords Cloud manufacturing · Software-defined networking
Cybersecurity · Manufacturing control

R. F. Babiceanu (✉) · R. Seker
Department of Electrical, Computer, Software, and Systems Engineering,
Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA
e-mail: babicear@erau.edu

R. Seker
e-mail: sekerr@erau.edu

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_18

1 Introduction

There is practically no industry domain left without being impacted by the advances in computing, and arguably by cloud computing. This tendency was seen in manufacturing operations for several years. Whether there are simple apps accessible from handheld devices, but which run in the cloud, or entire virtualized environments that provide real-time information on what happens in the physical processing world, cloud revolution is impacting manufacturing operations in a positive manner. It is expected that the current cloud operations trend will only be increasing in volume and complexity in the coming years. Also, it is expected that interoperability of cloud platforms and applications and security of electronic transactions will become an issue that require an adequate answer from cloud manufacturing actors.

Recent literature or research solicitations emphasize the need for securing the cyberspace in many domains. Manufacturing works with sensitive proprietary data, worth millions, so the security aspect of exchanging data over public-accessible networks cannot be overstated. Generally, using up-to-date communication standards and protocols, together with firewalls and other security measures, is good practice for securing manufacturing packet transmission. However, as it is well-known, the malevolent actors are continuously looking for ways to find any type of vulnerabilities that can be exploited.

This work proposes an interoperability framework for distributed manufacturing operations built based on software-defined networking principles. The proposed framework aims at assuring secure communication (packet transmission) over public-accessible networks by identifying appropriate communication standards and protocols and logical centralization of packet routing from one actor to another across the distributed interoperable manufacturing network. *Software-defined networking* (SDN) is a relatively recent networking approach in which the packet forwarding activity (also known as data plane or layer) is disassociated from the packet routing activity (also known as control plane or layer) and performed in a logically centralized fashion using SDN controllers in a remote location [1, 2]. This is opposite to traditional IP networks where the two activities are performed together within network routing devices (routers), where routing algorithms are locally run to identify the next path of the packets. SDN-based routing transforms networks routers in simple forwarding devices (called switches) which can be programmed remotely based on a centralized view of the network and any special requirements for the packets on the transmission path.

The SDN computing literature is abundant and SDN-based applications are already in use. For example, Google uses SDN for interconnecting its data centers, and CISCO and VMware provide application-oriented solution based on SDN packet routing [3]. However, SDN for manufacturing applications literature is rare and actual implementations are still only available in laboratory environments. There are just a few works that address the specific SDN needs for manufacturing applications [4–11], which still keeps the specific SDN manufacturing solutions part of an uncharted territory. This current work is viewed as a contribution to uncover solutions for aligning manufacturing applications to other SDN-rich domains.

From this point forward the paper is structured as follows: Sect. 2 presents the proposed SDN manufacturing interoperability framework together with its characteristics, and then, Sect. 3 details some of the technical aspects of the proposed framework through the use of a case study. Finally, the future research concerning the proposed manufacturing interoperability framework is outlined in the conclusions section.

2 SDN Manufacturing Interoperability Framework

2.1 Brief Review on SDN Models for Manufacturing Operations

Several general SDN survey works were identified in the literature [1, 2, 12–16], with some of them addressing the security-related aspects [1, 14, 15, 17–19]. The generic SDN principles extracted from the above works can be used in manufacturing applications, subject to their specificity. Interoperability frameworks can be found in several domains (e.g., civil aviation, healthcare providers, smart grid, etc.), in which there is a need to join together many different actors in a seamless interoperable type of network. However, as mentioned above, proposing a manufacturing-oriented distributed interoperability framework based on SDN principles is not that common; the few identified are summarized next.

A software defined cloud manufacturing architecture is proposed in [9], with a hardware plane and a software plane that includes virtual and control layers. Manufacturing resources are organized in workflows, which are included in the architecture as needed. The more and more present Industrial Internet-of-Things (IIoT), with its cyber-physical systems (CPS), industrial cloud, industrial networks, and public networks are all organized together using SDN principles, in what is called software-defined IIoT [10]. A prototype testbed implementation is also presented. The limitations of the traditional industrial networks (mostly wired connectivity, hierarchical structuring, offline pre-configuration, distributed network control and management, and lack of standardization) are outlined in [11]. The paper argues that the above limitations could be addressed with the adoption of application-oriented SDN, namely the proposed software defined industrial networks.

Some of the challenges of existing SDN solutions for industrial applications (plug-and-play for industrial manufacturing, deterministic real-time transmission services, mixed wired and wireless networks, and support for path redundancy for highly reliable transmissions), and proposed solutions to address them, are also reported [11]. In another work [6], an aggregation of CPS, that are part of a manufacturing systems environment, is considered. They transmit/receive production orders and implement production processes using network-based algorithms for routing time-sensitive communication flows.

2.2 The Need and Expected Benefits of SDN Manufacturing Interoperability Platforms

This current work builds on previous research [4, 5], where a testbed that implements the components of the SDN-based environments is considered. In addition, previous research discusses cybersecurity utilities and network resilience mechanisms included in the application level layer. The need for defining and implementing manufacturing interoperability platforms results from the challenges of current industrial networks, such as:

- Increasingly, industrial (manufacturing) networks are distributed in remote geographic locations, and must provide services subject to different rules, regulations, and cultural customs.
- Industrial (manufacturing) networks must link together different legacy or individual systems, not necessarily built for interoperability.
- Current industrial (manufacturing) networks have limited scalability and flexibility; new users, new applications, new data formats, and new packet size cannot be easily addressed.
- Many existing industrial (manufacturing) networks use proprietary communication and network management protocols, making the adoption of up-to-date security frameworks difficult, to say the least.

On the other hand, by moving the industrial (manufacturing) networks toward the SDN-based platforms, a series of benefits that address the above limitations is expected:

- Interoperability of systems composing the network.
- Improved and timely communication and decision making for all users.
- Support for adopted data formats and control of packets flow.
- Drastically improved network scalability and flexibility of operations.
- Implementation of security measures, and updates on a continuous basis.

2.3 SDN Manufacturing Interoperability Platform Components

The manufacturing application of SDN follows the three-plus-two-layer architecture, usually found in most of existing SDN-based implementations, also called

SDN reference model [12]. At the very bottom, the Data Layer includes the network hardware (switches) that forwards packets from one network node to another. This infrastructure could be composed of both proprietary and public forwarding devices. If public IP networks are used, they must include SDN-enabled switches. The Data Layer infrastructure extends horizontally to the manufacturing equipment and supporting cyber-physical devices. Next layer is the Southbound Interface, where the separation between the data and control plane takes place and routing paths are made available to the forwarding devices (switches) on the Data Layer. Information about the switches availability (failure or subject to attack) is sent to the Control Layer through this Southbound API. OpenFlow is the most used API standard for the interface, though others like OVSDB or OpFlex are mentioned in the literature [1, 20]. Figure 1 presents the SDN-based manufacturing interoperability platform architecture.

Then, the architecture includes the Control Layer, where the network operating system is located, and the packet routing is computed. Depending on the scale of the network, it could run on one server or a series of servers. Computation performed by the Control Layer includes also aspects related to network topology, bandwidth allocation, and any network monitoring function. The connection between the upper layer (Application Layer) and the Control Layer is made with the help of Northbound Interface. It is through this API that the application requirements (both network and manufacturing applications) are sent to the Control Layer for route computation based on specific packet types and security measures.

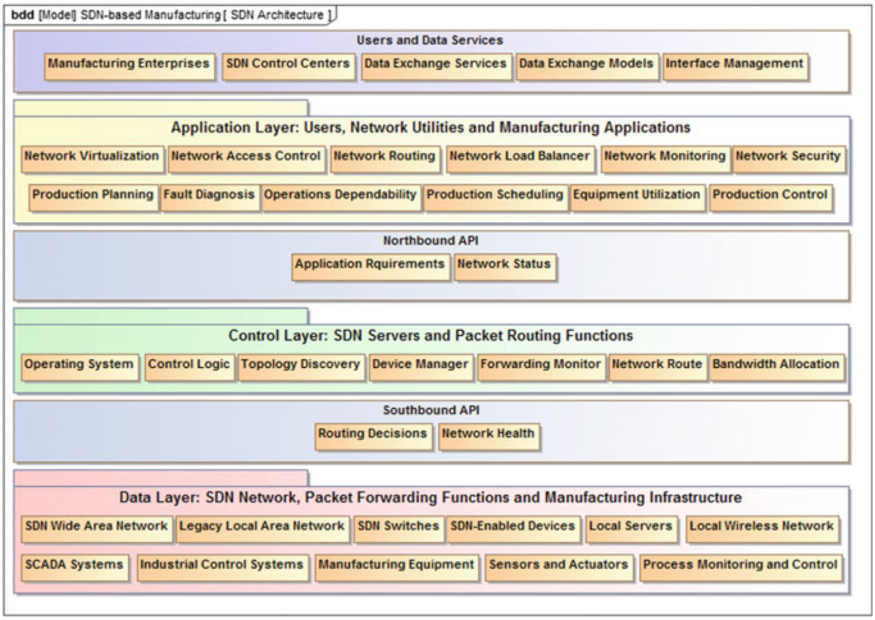


Fig. 1 SDN-based manufacturing interoperability platform

Network status information is also received through this API by the network utilities running on the Application Layer. There are several candidate solutions proposed by different vendors for this API. In many case, controllers come with their own APIs. The Application Layer is the upper layer where all the network applications are located. Network utilities, such as virtualization, access control, load balancing, monitoring, security implementation, and any domain-orientated applications are located and run here. On top of the Application Layer, the interoperability platform includes the actual users' access points, data exchange services, and interface management modules accessible by the users (usually manufacturing management/operations and SDN control centers).

3 SDN Manufacturing Interoperability Platform Performance

3.1 SDN Manufacturing Interoperability Case Study

This section investigates the performance of the SDN-based manufacturing environments in comparison with the traditional IP networks. While SDN software programmed networks can run on exiting IP networks if the network routers are SDN-enabled devices, pure SDN networks composed of SDN switches would be a better choice since the local router programming without a global network view is eliminated. A case study is considered to look into the communication transmission performance of two manufacturing enterprises remotely located and linked together through the SDN interoperability network. Each of the two manufacturing enterprises has several units that send and receive messages to/from the other. Once the messages are out of the manufacturing units they enter the SDN-based interoperability framework. The cases study is depicted graphically in Fig. 2. For the purpose of clarity, the model depicted in Fig. 2 includes only messages sent from the

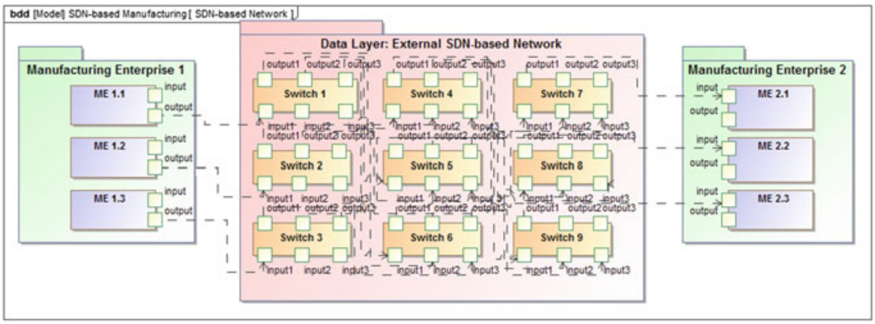


Fig. 2 SDN-based manufacturing network case study

first manufacturing enterprise to the second one. In operation, the SDN-based interoperability network would support communication in both directions.

3.2 Simulation Model for Manufacturing Interoperability

The case study uses an external network that is composed of SDN-enabled devices that can act both as a traditional IP network and as an SDN network. Using identical network topologies, two simulation models are considered. First model is using the traditional IP local routing mechanisms, while the second model is using the software programmed routing through logical centralization of network information, which is the SDN approach. The performance metric studied is the capability of the two models to forward data packets across the entire network when network nodes are subjected to Denial of Service (DoS) attacks. All the simulation variables such as packets created (communication load), packet processing at nodes (packet forwarding), and failure events occurrence (DoS attacks) come from the same statistical distributions, and use the same seeds for both models, such that the only difference between the two models is the routing decision: local IP routing vs. software-computed (SDN-based).

The actual network switches and bandwidth, switches ports, and network devices under attack for the two simulated configurations are depicted in Fig. 3. Once the DoS attack takes place, the network switch S4 is practically blocked from processing new legitimate packets (i.e., input ports s4.1, s4.2, and s4.4 cannot process incoming packets). Traditional IP network will address the "failure" of switch S4 by locally re-routing the packets that otherwise were choosing output ports s1.2, s2.2, and s3.2, to the other two output ports of switches S1, S2, and S3 (i.e., s1.3, s1.4, s2.3, s2.4, s3.3, and s3.4). First, the local routing uses Djikstra's shortest path and re-routes the packets to the closest network device. After receiving feedback related to network congestion, eventually, the local re-routing may be able to identify, after several updates of the forwarding tables, the network sections having larger bandwidth. On the other hand, given the logical centralization of the SDN network topology, the SDN-based routing is able to direct, from the beginning, the packets through the output ports that support larger network bandwidth (using switches S6 and S8).

The results of the two simulation models are presented in Figs. 4 and 5 for simulation replications started with the same seed. System performance is considered the amount of data packets forwarded in a unit of time. The more congested is the network, the less packets are forwarded, with potentially a part of them being dropped.

It can be inferred that the use of SDN networks makes the overall system performance better, which translates in a larger number of manufacturing data packets arriving at their destination in the time period under study. Also, while the network is under DoS attack, the percentage of dropped packets improves from 22.8% in the case of congested IP network, to 0.8% after congestion control is

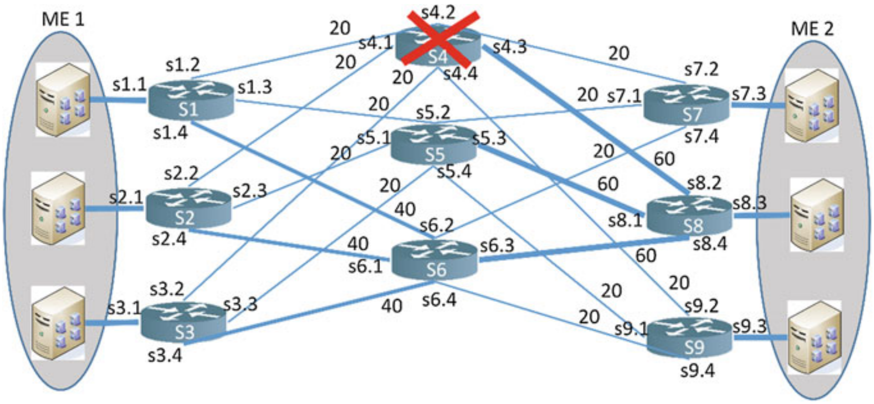


Fig. 3 SDN-based manufacturing simulation model

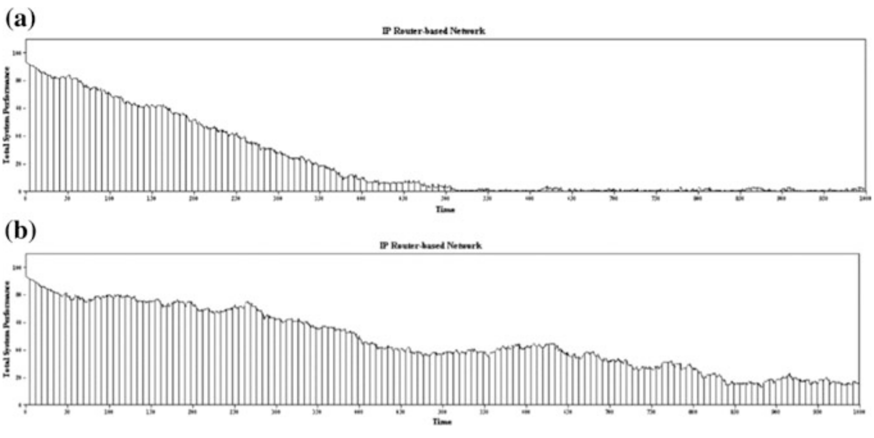


Fig. 4 IP network routing performance profile: **a** initial; **b** after congestion control

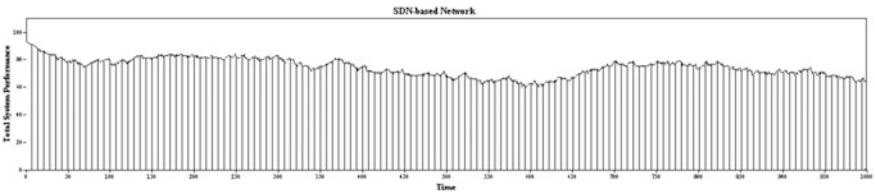


Fig. 5 SDN-based network routing performance profile

employed, and to only 0.3% in the case of SDN-based routing. The dropped packets need to be re-transmitted, which increases the average packet transmission time, thus further reducing the system performance.

4 Conclusions and Future Work Directions

This work proposes a SDN interoperability platform for manufacturing-based applications. Manufacturing must link together stand-alone resources, legacy industrial control systems, the new CPS and IoT, and modern shop-floor facilities. Different data format packets need to be transmitted, securely over high-speed networks where the potential for attack could be many times, considerably. Several other domains that, similarly to manufacturing, involve multiple users that exchange data across public or private networks already use, or are set to be using soon interoperability platforms. While there are SDN-based manufacturing models running in laboratory environments, the next years will, for sure, witness the move of manufacturing SDN from laboratory into practice.

The results of the small case study introduced in this paper shows that there is room for improvement in terms of the packet routing process in traditional IP networks, and SDN is one solution to address it. It is not appropriate for all public networking needs, such as regular Internet, but it would be particularly appropriate for manufacturing applications. Besides packet routing performance, SDN also improves the security of the packet transmission process, as the actual packet route is decided while having the total system view, rather than just immediate forwarding devices information. Any network device under question from the security point of view can be avoided using the software programming mechanism for routing data packets in SDN. There are also challenges regarding the SDN security, such as adding security layers to systems that run on older systems that were initially designed without security requirements in place, and with the goal of receiving very rare software changes/updates. On the other hand, cybersecurity needs frequent system updates. This is one of the most important aspects when the need to secure networks composed of older and newer systems arises, and is a research question to be researched in the near future. Other research questions, subject of the proposed work, are issues related to manufacturing resources network access, scalability, and data manipulation, which are all requirements for interoperable systems.

References

1. Kreutz, D., Ramos, F.M.V., Verissiomo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–75 (2015)
2. Jarraya, Y., Madi, T., Debbabi, M.: A survey and a layered taxonomy of software-defined networking. IEEE Commun. Survey Tutor. **16**(4), 1955–1982 (2014)
3. Morreale, P.A., Anderson, J.M.: Software Defined Networking Design and Deployment. RC Press Taylor & Francis Group, Boca Raton (2015)
4. Babiceanu, R.F., Seker, R.: Secure and resilient manufacturing operations inspired by software-defined networking. In: Borangiu, T., Thomas, A., Trentesaux, D. (eds.) Service Orientation in Holonic and Multi-Agent Manufacturing, SCI, vol. 640, pp. 285–294. Springer, Heidelberg (2016)

5. Babiceanu, R.F., Seker, R.: Cybersecurity and resilience modeling for software-defined networks-based manufacturing applications. In: Borangiu, T., Thomas, A., Trentesaux, D. (eds.) Service Orientation in Holonic and Multi-Agent Manufacturing, SCI, vol. 694, pp. 167–176. Springer, Heidelberg (2017)
6. Nayak, N.G., Durr, F., Rothermel, K.: Software-defined environment for reconfigurable manufacturing systems. In: Proceedings of the 5th International Conference on Internet of Things (2015)
7. Kim, D., Gil, J.-M.: Reliable and fault-tolerant software-defined network operations scheme for remote 3-D printing. J. Electron. Mat. **44**(3), 804–804 (2015)
8. Azaiez, S., Boc, M., Cudennec, L., Da Silva Simoes, M., Haupert, J., Kchir, S., Klinge, X., Labidi, W., Nahhal, K., Pfrommer, J., Schleipen, M., Schulz, C., Tortech, T.: Towards flexibility in future industrial manufacturing: a global framework for self-organization of production cells. Procedia Comput. Sci. **83**, 1268–1273 (2016)
9. Thames, L., Schaefer, D.: Software-defined cloud manufacturing for industry 4.0. Procedia CIRP **52**, 12–17 (2016)
10. Wan, J., Tang, S., Shu, Z., Li, D., Wang, S., Imran, M., Vasilakos, A.V.: Software-defined industrial internet of things in the context of industry 4.0. IEEE Syst. J. **16**(20), 7373–7380 (2016)
11. Li, D., Zhou, M.-T., Zeng, P., Yang, M., Zhang, Y., Yu, H.: Green and reliable software-defined networks. IEEE Commun. Mag. **54**(10), 30–37 (2016)
12. Xia, W., Wen, Y., Foh, C.H., Niyato, D., Xie, H.: A survey on software-defined networking. IEEE Commun. Survey Tutor. **17**(1), 27–51 (2015)
13. Jammal, M., Singh, T., Shami, A., Asal, R., Li, Y.: Software defined networking: state of the art and research. Comput. Netw. **72**, 74–98 (2014)
14. Yeganeh, S.H., Tootoonchian, A., Ganjali, Y.: On scalability of software-defined networking. IEEE Commun. Mag. **51**(2), 136–141 (2013)
15. Yoon, C., Park, T., Lee, S., Kang, H., Shin, S., Zhang, Z.: Enabling security functions with SDN: a feasibility study. Comput. Netw. **85**, 19–35 (2015)
16. Hakiri, A., Gokhale, A., Berthou, P., Schmidt, D.C., Gayraud, T.: Software-defined networking: challenges and research opportunities for future internet. Comput. Netw. **75**, 453–471 (2014)
17. Yan, Q., Yu, R.F.: Distributed denial of service attacks in software-defined networking with cloud computing. IEEE Commun. Mag. **53**(4), 52–59 (2015)
18. Wang, B., Zheng, Y., Lou, W., Hou, Y.T.: DoS attack protection in the era of cloud computing and software-defined networking. Comput. Netw. **81**, 308–319 (2015)
19. Ali, S.T., Sivaraman, V., Radford, A., Jha, S.: A survey of securing networks using software defined networking. IEEE Trans. Reliab. **64**(3), 1086–1097 (2015)
20. Kobayashi, M., Seetharaman, S., Parulkar, G., Appenzellar, G., Little, J., van Reijendam, J., Weissmann, P., McKeown, N.: Maturing openflow and software-defined networking through deployments. Comput. Netw. **61**, 151–175 (2014)

Manufacturing Systems at Scale with Big Data Streaming and Online Machine Learning

Octavian Morariu, Cristina Morariu, Theodor Borangiu and Silviu Răileanu

Abstract Real time analysis of data collected from the shop floor opens the path towards efficient scheduling of batch execution for large scale distributed manufacturing systems. Prediction of the shop floor activities has a great potential to reduce manufacturing costs, by providing the information required for operational decisions like preventive maintenance, automatic remediation or scheduling optimization. Research has been focusing on how machine learning algorithms can be used to better understand and extract insights from historical data collected from manufacturing systems. However, in the current manufacturing environments, driven by mass customization and short time to market, these approaches fail to be agile enough to be useful. In this paper we propose a real-time machine learning approach for large scale manufacturing systems that can predict various scenarios before service degradation occurs, thus allowing for corrective actions. At the same time, outliner detection algorithms can be used to evaluate the system's health at a holistic level. Scalability requirements are achieved by modelling the architecture around data streams processed in real time by map-reduce operations. The concepts presented in this paper build on recent developments on flexible, distributed and cloud based manufacturing, where these real time actions can be efficiently implemented.

Keywords Big data · Machine learning · MES · MSB

O. Morariu (✉) · C. Morariu · T. Borangiu · S. Răileanu
Department of Automation and Applied Informatics,
University Politehnica of Bucharest, Bucharest, Romania
e-mail: octavian.morariu@cimr.pub.ro

C. Morariu
e-mail: cristina.morariu@cimr.pub.ro

T. Borangiu
e-mail: theodor.borangiu@cimr.pub.ro

S. Răileanu
e-mail: silviu.raileanu@cimr.pub.ro

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_19

1 Introduction

Collecting real time monitoring data from shop floor devices is the first step in designing automated mechanisms capable of dynamic scheduling of manufacturing operations and handling unexpected situations, like faults in the shop floor resources. While totally unexpected breakdowns can occur and will unavoidably result in outages if those resources are not redundant in the system, a certain class of possible problems occurs over time. Consider a ball bearing as an example, the failure will not be immediate and slight variations in the system behaviour can be observed, like increased vibrations patterns or increased temperature in the affected component. In simple systems such changes in behaviour can be observed by human operators or by individual feedback loops on the corresponding metrics. A classical implementation would be to monitor the metric actively and set a fixed threshold for it; the threshold would be then used to trigger alerts.

In complex systems however, this approach is not efficient and does not work at scale. This is in part because not all faults can be predetermined or predefined; i.e. some vibrations can be caused by other factors that can be perfectly normal. On the other hand, some behaviour depends on the current system context that, in complex systems, can have virtually an infinite number of states. These include the products being manufactured on the shop floor in that given time interval, the interactions between shop floor resources, material and information flow. Another important consideration is the ability to track and consider the covariance of multiple metrics monitored. Continuing on the previous example, if a sensor monitors vibration while another one heat, the covariance of these two might indicate the difference between a fault and normal behaviour. Again, this covariance might be difficult to define ahead of time, as interactions between components can be very complex and almost always change in time.

In this context, we propose that an efficient manufacturing execution system (MES) must have the following functional characteristics:

- Dynamic scheduling of manufacturing operations based on real time data and predictions derived from it for the near future;
- Dynamically learn the patterns of the signals monitored;
- Detect faults before service degradation occurs;
- Determine and learn the covariance between signals;
- Work on real time data streams from sensors rather than static data;
- Classify the current state of the manufacturing system as healthy or faulty;
- Execute automated corrective actions.

Some qualitative considerations are also required, especially related to the number of false positives the system generates. While in a completely accurate system it won't be feasible, keeping the false positives rate low is usually the decisive factor when considering automating the corrective actions. As the time interval between detecting a possible fault and when service degradation starts occurring can be low, automating the corrective actions is a fundamental feature of a dynamic MES system. One possible solution for minimizing false positives is to provide the system with the ability to learn from the outcome of its own actions.

In this paper we propose a MES system designed for dynamic scheduling of manufacturing operations using map-reduce style aggregations of real time data streams from shop floor. Machine learning is used for short term prediction of some key performance indicators (KPIs), specifically linear regression for scalar predictions, and k-means for classification problems. The pilot implementation is using Apache Kafka and Apache Spark application stack; the most important technical aspects of the architecture are discussed in Sect. 3.

2 Related Work

Real time decision for online operation scheduling is currently a popular research topic. Some initial steps towards real time data integration in MES systems were presented by Zhong et al. [1] where RFID technology was used to track the movement of objects in the shop floor. In this approach, RFID devices are deployed systematically on the shop-floor to track and trace manufacturing objects and collect real-time production data. However, the process described is mostly manual and the decisions are taken by human operators. This does provide improvement compared to the previous hierarchical MES models, but it does have scalability limitations and is error prone due to the human factor involved.

In [2], Zhang proposes a real-time, data-driven solution that optimizes decision using a dynamic optimization model. The model is based on game theory and one of the key differences is that each machine is an active entity that will request the processing tasks independently. While this is an important step towards decentralization, the solution does not include machine learning techniques in order to have a predictive behaviour. While real time data is required for this class of MES implementations, a predictive horizon can be established in combination with machine learning, which adds a new layer of possible optimizations.

On the machine learning side, He [3] proposes a KNN (k nearest neighbour) approach to detect faults in the semiconductor manufacturing process. This approach is interesting in the fact that it covers the case of multivariate analysis in the context of mixed product batch. However, it does not correlate the shop floor resource metrics, so it fails to predict resource faults before system degradation occurs.

Another interesting approach is using Petri nets [4] to model the fault detection system and subsystems. While the approach is flexible in nature and can be reconfigured for various layouts, it lacks the ability to learn from the environment itself. Also, the hierarchical system design assumed might not apply directly to distributed manufacturing systems. Ideally, a fault detection system must consider all sensor data streams without being aware of the actual layout. At the same time, the covariance or the relative relations between sensors should be automatically detected.

Artificial Neural Networks (ANNs) are used in [5] to construct and evaluate a fault detection system against a rule based approach. The ANN implementation has the ability to determine faulty states by multivariate classification. However, it cannot detect faults or anomalies that occur at distinct time intervals, for example, if sensor_1 shows a fault a time T1 while sensor_2 shows a fault at T2. These two faults can have an important covariance if they occur in the same time window.

This paper tries to bridge the gap between previous real time MES implementations and the new machine learning techniques that can be used to predict future behaviour on the shop floor. Machine learning is, in general terms, a very powerful tool to extract insights from data. Traditionally, machine learning has been used on static or historical data. However, if the data can be obtained in real time and the machine learning algorithm can be run in a real time context with re-training on new data, then the insights become predictions, enabling real time decisions.

3 System Architecture

The architecture uses the manufacturing service bus (MSB) to collect the real time metrics and events from the shop floor devices. While this paper does not attempt to focus on a specific shop floor/MES design, it can be considered that the shop floor is consisting of resources capable of reporting their state and a set of KPIs, and intelligent products too. The information flow creates at high level a loop, such that real time data streams are flowing from the shop floor into MSB; from there, a series of map-reduce aggregations are performed, followed by machine learning on aggregated streams. Once the information is correlated and consolidated, the scheduler uses that information to determine the next operations to be performed and their location. These scheduling decisions are passed back to the shop floor again through MSB.

Figure 1 illustrates the high level architecture. In the reminder of this section, the individual components will be discussed in detail.

3.1 Manufacturing Service Bus

Working with real time data raises several practical problems that have to be addressed at design stage. Firstly, it is usually impossible to know in advance the amount of data that is generated by each shop floor resource, or the rate at which this data is being sent. This introduces the need for a robust queuing system, capable of handling data bursts in a consistent manner. Such a system—the Manufacturing Service Bus (MSB) is presented in [6] and has the following communication characteristics (Fig. 2):

Event driven communication: at shop floor level there are a high number of events generated that need to be dispatched to the upper level. For example, when a

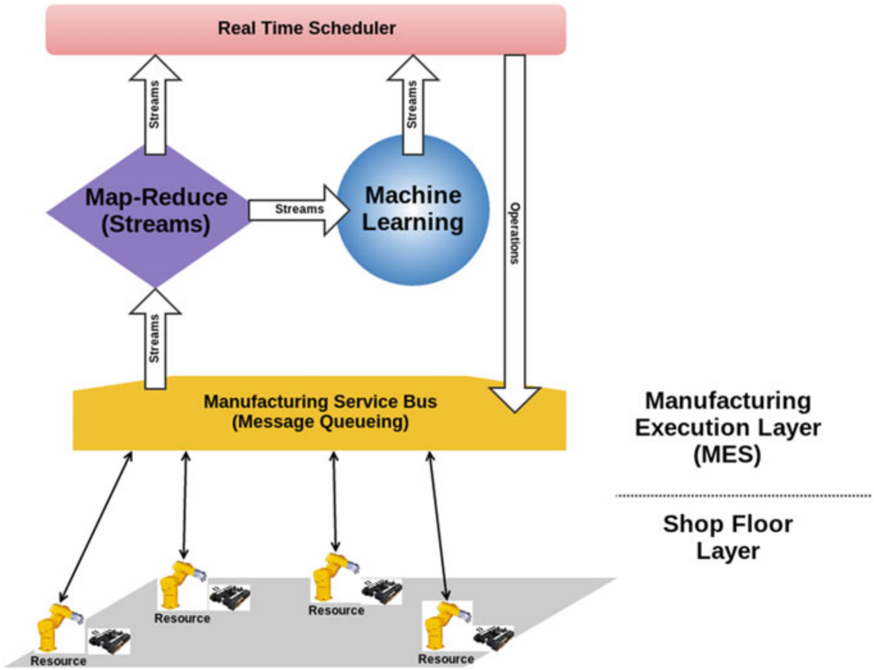


Fig. 1 Real time MES architecture with integrated machine learning

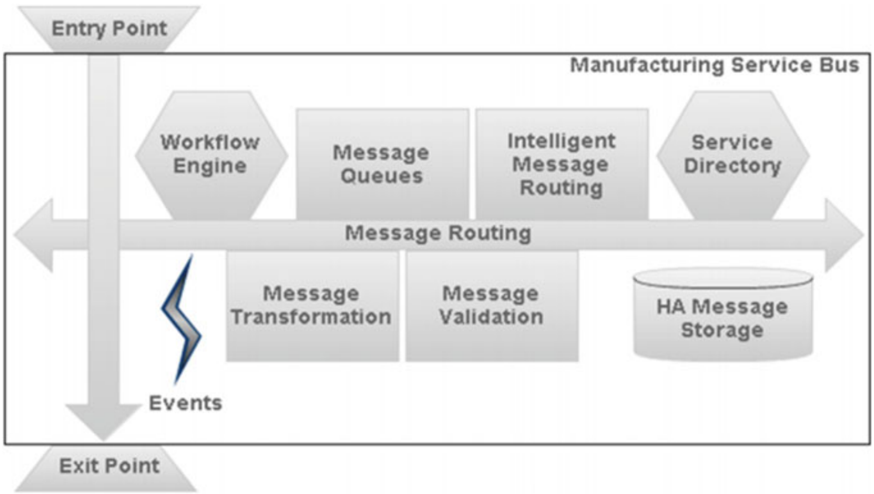


Fig. 2 Manufacturing service bus logical view

pallet arrives in a given position on the conveyor belt, a sensor detects the associated RFID tag and generates an event. This event needs to be dispatched to the relevant resources in order to be processed by the scheduling module. The events are routed via event streams, in which their chronological order is preserved.

High number of messages: the MSB is a very effective architecture providing that all the other components are using it to exchange messages with each other, or in other words, the MSB is not bypassed by using direct point to point communication. However, the disadvantage is that in complex manufacturing systems the number of messages passed can grow exponentially based on the number of modules involved and the number of products. From a MSB implementation perspective it is important to assure that a high message throughput is possible. This requirement is the basis for scaling of communication platform for real time data.

Message transformation: the shop floor level integrates a wide range of modules, from software schedulers to various hardware devices (robots, sensors, etc.). From a communication perspective, the protocols and the message formats used can be a simple +5V DC signal, proprietary line protocols or event high level TCP based protocols. The role of MSB is to transform these messages to and from these proprietary protocols in a common standardized format. This is done by allowing the development of message convertors at any entry point and exit point of the bus. Message convertors assure that data streams can be aggregated at upper layers by enhancing the messages with relevant metadata (ID of the shop floor resource, location, production batch ID, etc.).

Message validation: the MSB is the appropriate place, from an architectural point of view, to perform message validation. Malfunctions or defects in shop floor devices can generate messages that are invalid. The best practice is to validate according to predefined rules and logic each message before dispatching it through the MSB. This approach prevents errors down the stream in the map-reduce procedure, which can be costly from a computational point of view.

Synchronous and asynchronous communication: the MSB implementation offers both synchronous and asynchronous communication models. The synchronous model causes the sender of the message to block until the response is received and so is implicitly bidirectional. The asynchronous model is using a queue-based mechanism, where the sender submits the message and from where the receiver picks it up at a later time. This allows decoupling of the execution of the sender and the receiver. At the shop floor level both communication models are useful.

Message persistence: when the asynchronous model is used, the messages reside in logical queues from where they are consumed. The MSB implementation stores the queues in a persistent highly available storage that allows production state recovery in case of a system crash. The MSB can use a network file system or a distributed database as a repository for the message queues.

Distributed execution: the IT infrastructure at the shop floor level is distributed along several devices: resource workstations, controllers, servers and even embedded devices travelling on pallets in the production line. The MSB implementation runs in a distributed model as well, allowing access to the MSB functionality for all devices.

3.2 Streams and Map-Reduce Engine

Map-reduce [7] is a programming model suited to resolve a great variety of big data problems. The concept relays on programming two functions: a *map* and a *reduce* function. To put it in simple terms, the map function has the role of transforming a data point into a key/value pair. The reduce function applies an operation on the values of the same key. While the map-reduce programmer is concerned only with the implementation of these two functions, the actual implementation [8–10] takes care of distributed execution, resource allocation, failures, and results aggregation. This approach allows for highly efficient use of computing resources, and thus allows almost horizontal scaling of the system. It is worth noting that initial implementations like Hadoop are focusing on static data, while the recent trend is towards stream processing (Apache Spark and Apache Flink projects).

In the context of the real time MES system proposed in this paper, the stream processing is a compulsory feature. We propose the following simplified stream structures passing via the MSB, as shown in Fig. 3.

Resource Streaming: Shop floor resources typically send data encapsulated in events. These events can be periodic or as a response to an unexpected situation, like resource breakdown or stock depletion. The periodic events encapsulate monitoring data, including operational parameters (various KPI values, like average power consumption, temperature, vibration levels, etc.) and functional parameters like time taken for a given operation or current material stock. Each of these events has a timestamp in order to assure valid time series streaming.

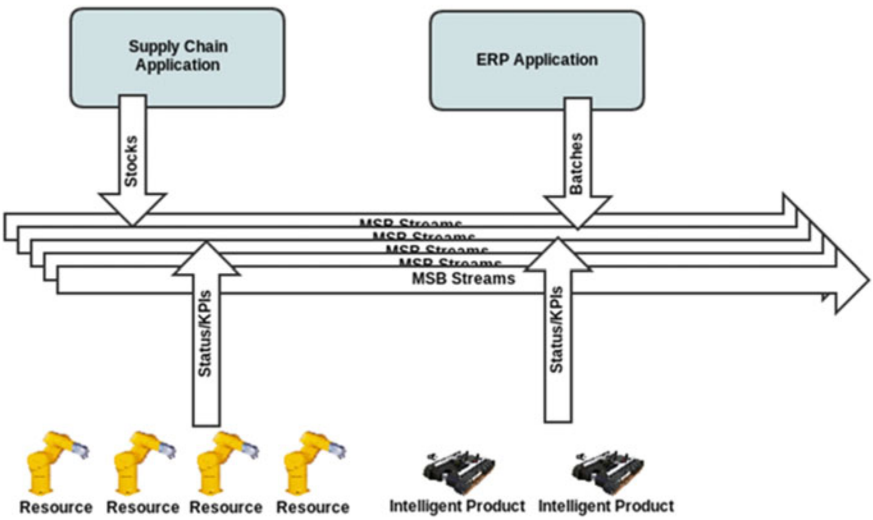


Fig. 3 Simplified stream structure

IP Streaming: Intelligent products data streams contain mostly location and status information. The place where the product is located on the shop floor is a relevant information for scheduling and path optimization.

Along with the shop floor streams, there can be multiple upper layer streams coming from Supply Chain applications that can indicate real time stock availability and Enterprise Resource Planning applications, that contain batching information and possible disturbances in the production process like rush orders. The above are presented as examples in order to illustrate the aggregation techniques; in practice the nature of these streams will depend highly on the enterprise layout and internal information flow. One map-reduce model that can be used, is based on aggregating the data based on production batch ID as illustrated in Fig. 4.

If a similar map-reduce approach is applied on each data stream, and then the resulting streams are merged and reduced again, a consolidated data stream is

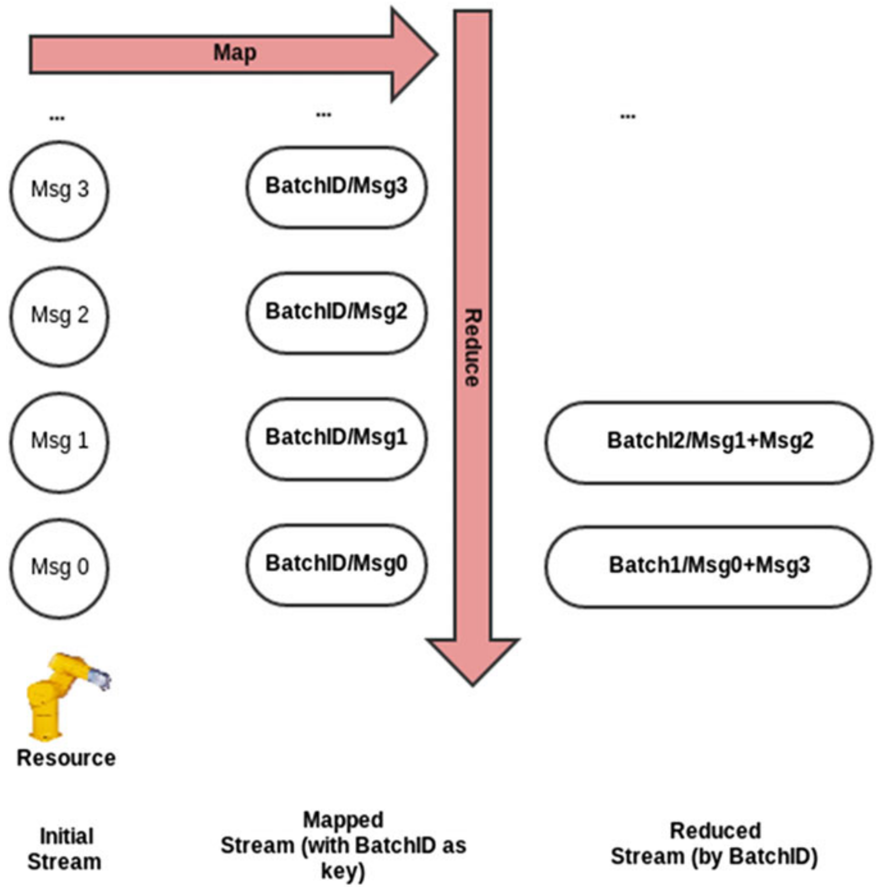


Fig. 4 Map-reduce on resource stream based on batch ID

obtained that has all the merged information from all resources, and all products involved in a particular batch ID at a given time. Note that this aggregation function is executed by the map-reduce engine in distributed fashion on multiple compute nodes. This assures the horizontal scalability of the solution.

In practice, multiple aggregations can be computed, by using different keys for map-reduce. One interesting approach is to tag the messages by their physical location in the shop floor. Intuitively, this will result in having a consolidated stream that shows in real time the complete information for a location (resource status merged with close by products and available local stock). Similarly, other streams can be defined on energy consumption areas or heat distribution.

Another important consideration is that the map-reduce function is applied in batch mode, and typically the batch is time-based. In practice this means that the map-reduce will be executed on repetitive basis, several seconds apart, and the portion of the batch relevant for that time series will be included in the operation. The interval of the map-reduce execution depends on how quickly the data should pass through the system.

3.3 Machine Learning and Predictions

Once consolidated streams are available at the output of the map-reduce algorithm, the data can be directly used for decision making. For example a scheduler implementation can consider the energy consumption mapped on resources in order to schedule the next operation such that energy consumption is optimized. Another scheduling strategy might be to choose the closest free resource, if location is used as a key.

However, in some cases, it is useful to attempt predicting some behaviour in the near future, and use that information for scheduling operations. While machine learning is a wide research field and there are many algorithms that can be used, in this paper two main approaches are presented that cover some of the basic use cases.

Firstly, when dealing with scalar data, for example energy consumption for a given operation from a resource, one could use linear regression to see if there is a pattern in the data. To do this, the energy consumption for a given operation can be map-reduced and aligned chronologically in time. Once this is done, linear regression can be applied to the time series and a threshold can be used to check the pattern.

Figure 5 illustrates this approach. On the horizontal axes we have the timeline (seconds to current time) and on the vertical axes we have the recorded power consumption. Applying linear regression on this micro-batch would result in determining the slope of the line in the right graph. This allows for *prediction* of how much energy would that operation require in the next runs, at least in a small time horizon; but as data and scheduling are real time, even a small horizon can lead to significant optimization. At the same time, by applying a threshold on the slope,

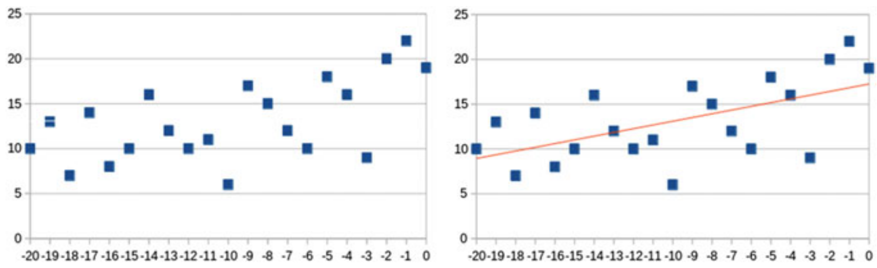


Fig. 5 Linear regression applied within a stream

the scheduler can detect if the resource allocation preference should be changed to optimize energy consumption. Similar approaches can be applied to other streams, based on operation duration for example. It is important to note that these predictions can be made in parallel, and the scheduling algorithm can use multiple depending on the local optimization goals.

Another machine learning technique that can be applied at this layer is *anomaly detection*. The classification can be done at multiple levels. For example, one classification can be the combined system health, considering multiple combined KPIs in the map-reduce phase. Once all KPIs are available in a consolidated stream, Principal Component Analysis (PCA) can be applied to reduce the dimensions of the problem from N (total number of KPIs considered) to two. Then, an anomaly detection algorithm can be applied to classify the system state. In this case, a binary classification of the system state is considered, where the MES system can be either in a healthy state or in an unhealthy state. In this context our goal becomes in identifying outliner states in the data. Essentially if the system state, after PCA dimensional reduction, is further away from the centre of the good known states, it is considered as an anomaly. This in turn triggers corrective actions. Figure 6 illustrates this algorithm applied to a rolling set or 50 system states. As shown in the figure, two states have been detected as outliners. When such a state is detected in real time, corrective actions can be applied.

It is important to note, that in the context of a real time MES system, the states considered for outliner detection can be just part of the current map-reduce batch or can contain a rolling baseline of historical system states. In practice, this depends on how the states space changes in time, and how adaptive the system should be. If we have a relatively stable system, with similar characteristics over time and products being manufactured, a long historical baseline of system states will give better results. However, for a dynamic system, a short baseline would allow for an adaptive behaviour.

Similar approaches for outliner detection can be used in smaller scope as well. For example it can be applied to a group of shop floor resources, or even individual resources. The results can then be consumed by the scheduler service in real time to optimize the resource allocation and operation sequencing.

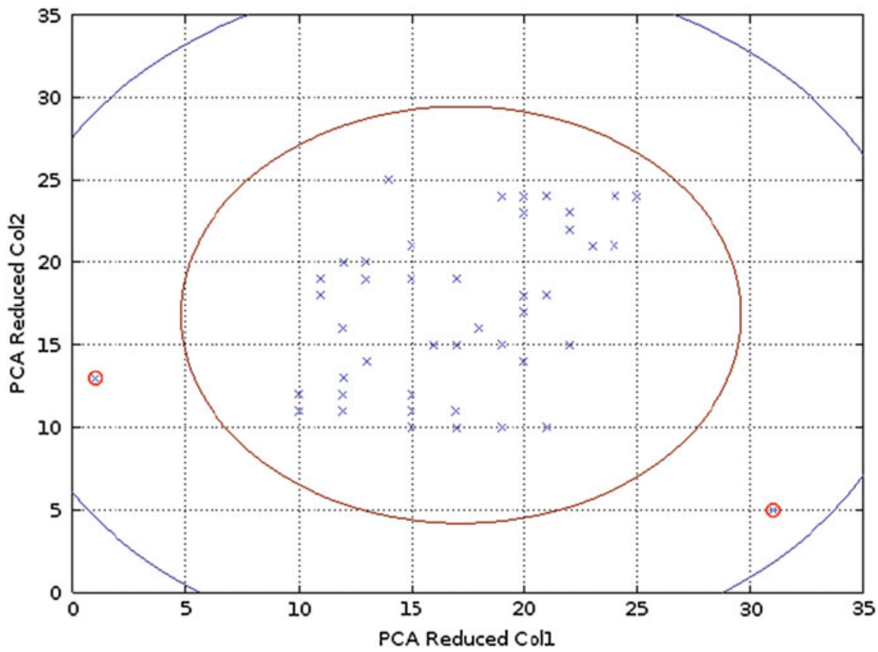


Fig. 6 Anomaly detection for MES systems after PCA reduction (axes have no specific meaning after PCA)

A pilot implementation is currently being developed using Apache Kafka as a platform for MSB implementation and Apache Spark for map-reduce stream processing. Spark ML module provides convenient implementations of linear regression and k-means classification algorithms that can be directly used on streams.

4 Conclusions and Future Work

Machine learning techniques have the potential of unlocking a new layer of possible optimizations in MES systems. However, as discussed in this paper, the effectiveness of machine learning depends on the real time aspect of the data used in training. The results of these algorithms are often accurate in a small time horizon in the future, so only a real time scheduling engine can benefit from these approaches. There are two main use cases for MES systems: predicting values of scalar KPIs and making scheduling decisions on them, and outliner detection.

On another hand, real time machine learning requires online, on the fly training regardless of the algorithm. This training is resource intensive, thus a map-reduce architecture becomes mandatory, assuring distributed execution and horizontal scalability. Once these two concepts are combined, it opens the way for new MES designs that act in real time on predicted data, thus anticipating the shop floor events and allowing optimization and prevention of failures.

Future work is focused on implementing a reference system where various use cases can be identified, tested and benchmarked, offering a platform for evaluating both map-reduce implementations and machine learning algorithms along with the new scheduling mechanisms capable of consuming this predictive data. Fault detection mechanisms can also benefit from this approach, as deciding on corrective actions before service degradation occurs can help in preventing outages.

Acknowledgements This research work has been partially supported by the IBM FA 2016 project: Big Data, Analytics and Cloud for Digital Transformation on Manufacturing—DTM, period of execution 2016–2018.

References

1. Zhong, R.Y., et al.: RFID-enabled real-time manufacturing execution system for mass-customization production. Robot. Comput. Integr. Manuf. **29.2**, 283–292 (2013)
2. Zhang, Y., et al.: Game theory based real-time shop floor scheduling strategy and method for cloud manufacturing. Int. J. Intell. Syst. (2017)
3. He, Q.P., Wang, J.: Fault detection using the k-nearest neighbour rule for semiconductor manufacturing processes. IEEE Trans. Semicond. Manuf. **20.4**, 345–354 (2007)
4. Viswanadham, N., Johnson, T.L.: Fault detection and diagnosis of automated manufacturing systems. In: Proceedings of the 27th IEEE Conference on, Decision and Control. IEEE (1998)
5. Heshan, F., Surgenor, B.: An unsupervised artificial neural network versus a rule-based approach for fault detection and identification in an automated assembly machine. Robot. Comput. Integr. Manuf. **43**, 79–88 (2017)
6. Morariu, C., Morariu, O., Borangiu, T.: Manufacturing service bus integration model for implementing highly flexible and scalable manufacturing systems. IFAC Proceedings Volumes **45.6**, 1850–1855 (2012)
7. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51.1**, 107–113 (2008)
8. http://hadoop.apache.org/
9. http://spark.apache.org/
10. https://flink.apache.org/

Part V
Simulation for Physical Internet and Intelligent and Sustainable Logistics Systems

Proposition of an Implementation Framework Enabling Benchmarking of Holonic Manufacturing Systems

Olivier Cardin and Anne L'Anton

Abstract Performing an overview of the benchmarking initiatives oriented towards the performance evaluation of Holonic Manufacturing Systems shows that there are very few of them. However, a comparison between all the isolated emulation developments for benchmarking in literature was made, and showed that many common features could be extracted. Several deadlocks for a generic approach of these developments are also exhibited. A global architecture dedicated to generic performance evaluation platform design is suggested. This architecture integrates a scenario manager, whose main specificities are detailed and justified. These features serve to both integrate the best practices encountered in literature and fulfil the missing aspects to respond to the problematic.

Keywords Virtual commissioning · Emulation · Performance evaluation Benchmarking · Simulation

1 Introduction

Current research and developments in next generation manufacturing control systems, and specifically Holonic Manufacturing Systems, recently emphasized the maturity of the underlying concepts and methods [1]. In this context, the next step is dissemination of concepts, primarily through a wide industrial acceptance of the related developments. However, these control architectures suffer from a lack of performance guarantee, as they are mainly based on emerging behaviour techniques, such as multi-agent systems or holonic paradigm, making the performance of the control system highly dependent on the context of execution of the experiment [2].

O. Cardin (✉) · A. L'Anton
Laboratoire des Sciences du Numérique de Nantes, LS2N UMR CNRS, LUNAM Université, IUT de Nantes – Université de Nantes, 2 avenue du Prof. Jean Rouxel Carquefou, Nantes, France
e-mail: olivier.cardin@ls2n.fr; olivier.cardin@univ-nantes.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_20

Traditional benchmarking activity consists in evaluating the response of the control system to a manufacturing situation with a predefined set of data. Several years ago, the operational research (OR) community has proposed several benchmarks to try and compare the algorithms solving static NP-hard optimization problems for production, among which [3] is one of the first. This approach is not fully satisfying for next generation control systems. Indeed, their major interest relies in their robustness and reconfiguration abilities that requires to be evaluated online [4]. Therefore, a whole new evaluation framework includes both the final control system plugged on an emulation/virtual representation of the manufacturing system in one of the scenarios called High Level Virtual Commissioning expressed in [5]. A performance evaluation conceptual framework was developed for assessing the level of quality of a scheduling solution in terms of efficiency, robustness and flexibility [6], and defined several years ago the general architecture of an online benchmarking instance (Fig. 1) which exhibits perfectly the full decomposition between control system and emulation.

The main problematic the community is currently facing is the lack of details of this generic framework, making each application developed ad hoc with various functionalities and possibilities. The aim of this paper is to suggest an implementation framework of both the control system and the emulation model in order to standardize the development of such initiative and allow the application of various benchmarks.

Therefore, a comprehensive analysis of the existing benchmarks in literature is performed in the second section in order to exhibit the requirements for the range of scenarios to incorporate in the framework. Then, a review of some of the existing emulation models developed in literature is proposed in the third section in order to emphasize the convergence between each individual initiative. Finally, the resulting framework is presented in the fourth section.

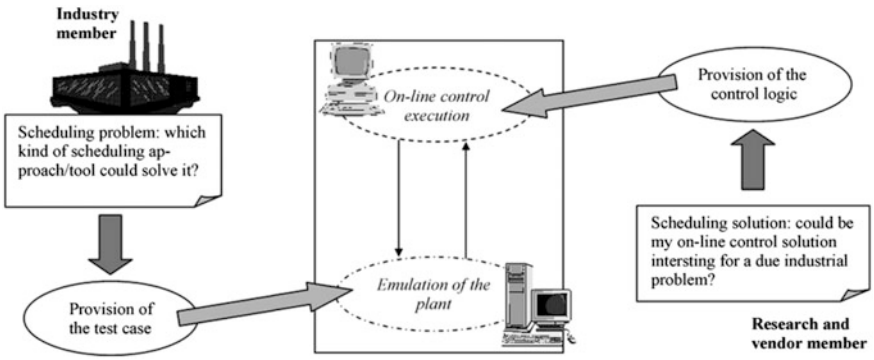


Fig. 1 General framework of an online benchmarking instance [6]

2 Benchmarking Holonic Manufacturing Systems

Evaluating the performance of Holonic Manufacturing Systems is reputed to be a difficult task, as it requires a dynamic evaluation of the control system's response to predefined scenarios, as much as a prerequisite for industrial acceptance. As a matter of fact, numerous works in literature can be found that exhibit a performance evaluation, but generally on ad hoc scenarios fully customised for the dedicated application. Among these, Jovanovic et al. [7] studies the implementation of a holonic control system on a "green"-tyre-manufacturing system. The objective is to evaluate how the holonic control is able to eliminate the impact of machine breakdowns on productivity. To do so, two examples of scenario are chosen, and the comparison with so called classical control approaches exhibits a 4% increase of productivity.

Table 1 synthesizes, for each of the initiatives presented in this section, the type of scenarios encountered. It states for each of the benchmarks encountered in literature (in columns) the category of scenarios that are taken into account. These categories are:

- Dynamic reconfiguration, impacting the whole system, typically machine failures;
- Quality issues, impacting mainly the products, typically rejection or remanufacturing of products;
- Order management, impacting the control system, typically cancelled or high priority orders;
- Supply issue, impacting the control system, typically a shortage in components on a machine.

Table 1 Holonic Manufacturing Systems benchmarking scenarios categorization

Category	Directly impacted element	Unstable conditions [7]	Manufacturing disturbance scenarios [8]	Experimental modalities [9]	Dynamic production system scenarios [4]
Dynamic reconfiguration	Shop floor resources	Example 1; Example 2	Query 2; Query 4; Query 5; Query 6; Query 9; Query 10	PD1; PD2	#PS2; #PS3; #PS3; #PS7; #PS9; #PS10; #PS12
Quality issues	Products				#PS6; #PS11
Order management	Control system		Query 3; Query 7; Query 8	BD1; BD2	#PS1; #PS4; #PS13; #PS14; #PS15
Supply issues	Control system				#PS8

Bal and Hashemipour [8] suggest a virtual reality-based methodology for enhancing the design and implementation process of holonic control systems in manufacturing practice with the objective of implementing and disseminating holonic control into the small to medium size manufacturing enterprises. The case study is developed on a die-casting factory, *Sahin Metal*, in Istanbul, Turkey. The objective is to measure Throughputs, Lead Times and Resources utilization considering 10 different scenarios, called "Manufacturing disturbance scenarios". Those scenarios extend the range of considered cases in the following directions:

- The reconfiguration of the system also consider rapid insertion of new resources or modification of their capabilities;
- The information system is considered, with the management of the order and their dynamic evolution (rush orders for example—see Table 1).

Even if Table 1 exhibits a relative convergence of the considered scenarios, initiatives tried to define a full methodology to design the benchmarking experiment. Among those, Mönch [10] suggested the following scheme in order to construct the benchmark:

1. Determination of production control approaches used for comparison;
2. Determination and specification of the used performance measures;
3. Specification of the used performance assessment strategy;
4. Description of the hardware and software environment for the benchmark;
5. Description of different scenarios that should be simulated; this includes especially the description of designed experiments;
6. Simulation of the scenarios and discussion of the results.

Pannequin et al. [9] defined a benchmarking protocol, targeting HMS implementation projects. A component-based generic architecture is proposed with this protocol, enabling to model and compare various control architectures. The case study relies on an automotive-industry. Business oriented disturbances (BD) are considered (Order management) along with Process oriented disturbances (PD) that relate to Dynamic reconfiguration.

Finally, the Bench4Star initiative [4] is probably currently the most advanced benchmark for HMS in literature. As exhibited in Table 1, more scenarios are taken into account with Quality and Supply issues, which extend the range of the evaluations and make the scenarios closer to real manufacturing conditions.

3 Emulation of HMS

3.1 Development Approaches

From the individual initiatives that were developed among the years in literature, an empiric approach in the development of emulation-based performance evaluation of

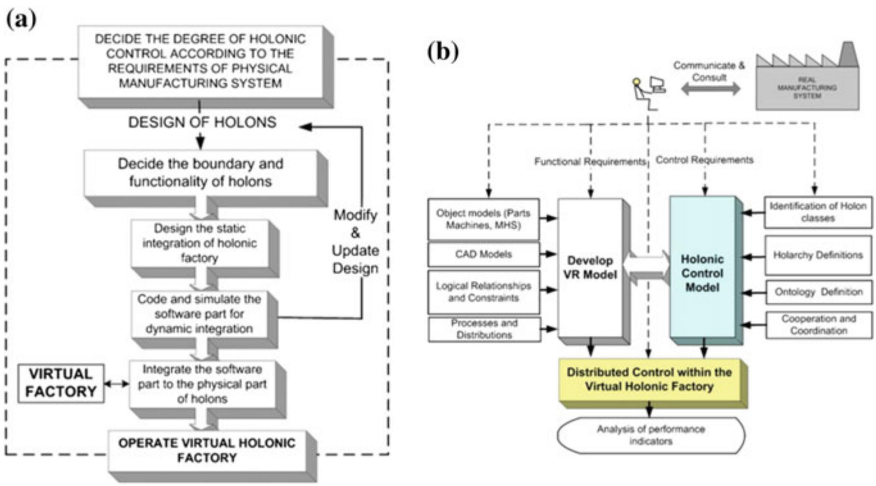


Fig. 2 Design approach of emulation-based performance evaluation (**a**) and development detail of emulation architecture (**b**) [8]

HMS control was designed (Fig. 2) [8]. In the general approach (a), the emulation issue is mainly located in the bottom part. Part (b) of Fig. 2 represents with more details the development process of the virtual factory model. Two elements might be noticed. First, the scenarios are not mentioned, which implies the necessity to develop ad hoc models for each tested scenario. Second, the VR model returns performance indicators for the analysis of the response of the control to the scenario.

In the same way, a software architecture was suggested by [10] (Fig. 3). It was designed for a full integration with C ++-based control system and Java simulation tools, and a web-based access to allow the users building their own simulation models from scratch by specifying the simulation model in XML format. In this architecture, the coupling between the control algorithm and the emulation is loose, and the performance indicators are extracted both from the emulation and the control system.

The analysis of both these approaches exhibit several problems:

- #P1: What is the coupling for a generic approach between holons and emulation?
- #P2: How to retrieve the performance indicators?
- #P3: Which integration of the scenarios in the architecture is possible?

The next section performs a literature review of the proposed developments and examines their response to these questions.

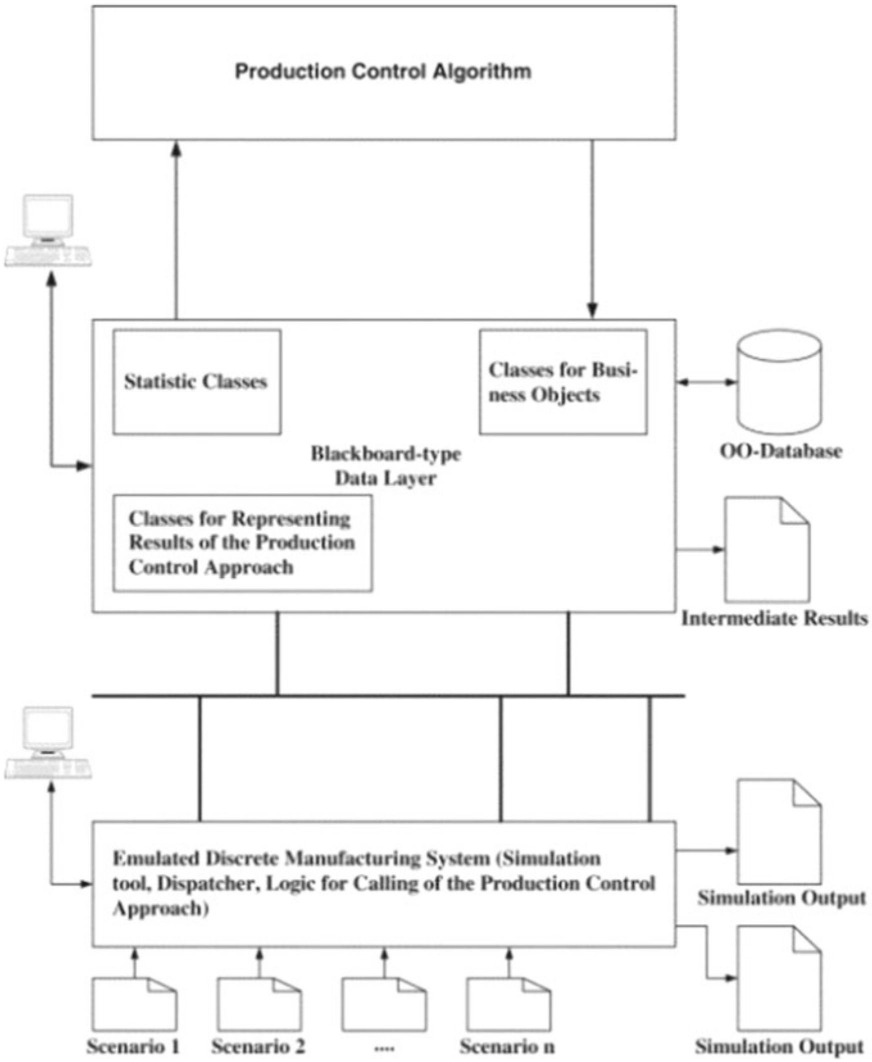


Fig. 3 Software architecture of emulation-based control [10]

3.2 HMS Emulation Literature Review

3.2.1 Answers to #P1: Coupling HMS/Emulation

Several studies were lead on the genericity of the approach of emulation, such as [9] or [10] that were previously mentioned. Another interesting initiative was called Arezzo-FMS [5]. The idea was to develop a generic emulation model and

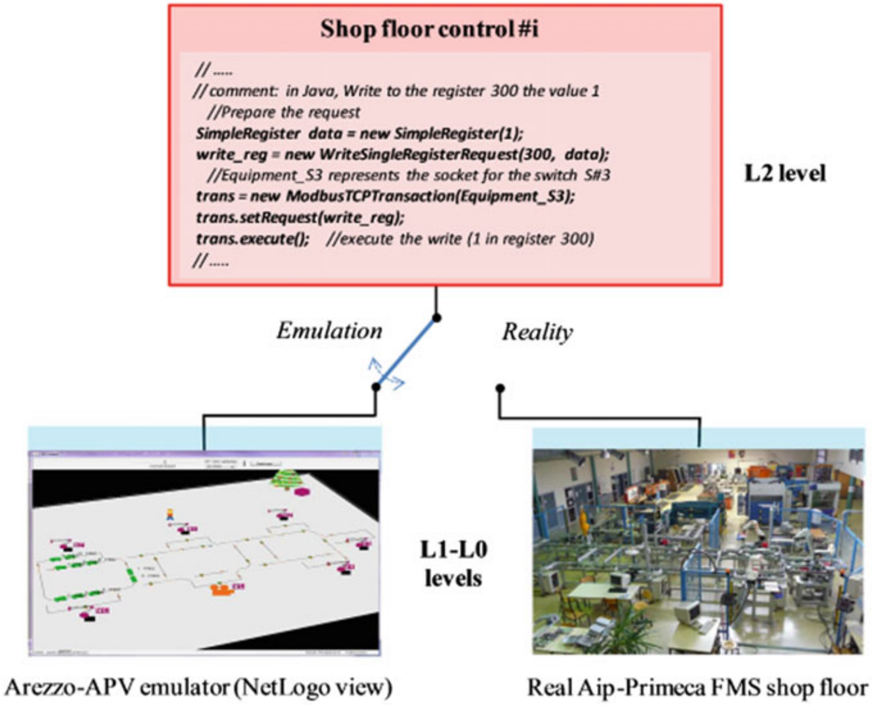


Fig. 4 Arezzo-FMS general objective [5]

generation methodology that was able to connect to the shop floor control, which is easily reconnected to the real shop floor (Fig. 4).

In this context, they introduced the concept of *Interface Layer (IL)*, which is one of the major development primitives, allowing the communication between holons and simulated objects the same way they do with real shop floor entities (Fig. 5). Examining the various studies that also exhibited the use of emulation for performance evaluation, this point is frequently dealt with (Fig. 6).

The three studies exhibited in Fig. 6 have various purposes: case (a) is related to the holonic control in tyre manufacturing industry [7], case (b) deals with the control of a flexible manufacturing system [11], whereas case (c) intends to validate the behaviour of a holonic controller of modular conveyor systems [12]. They were developed in parallel without interaction, but show several common features. One of them is the presence of the IL at the interface between the virtual model and the real control to be tested. Identical conclusions can be drawn about cloud simulation platforms [13] or agent-based manufacturing systems [14], for example.

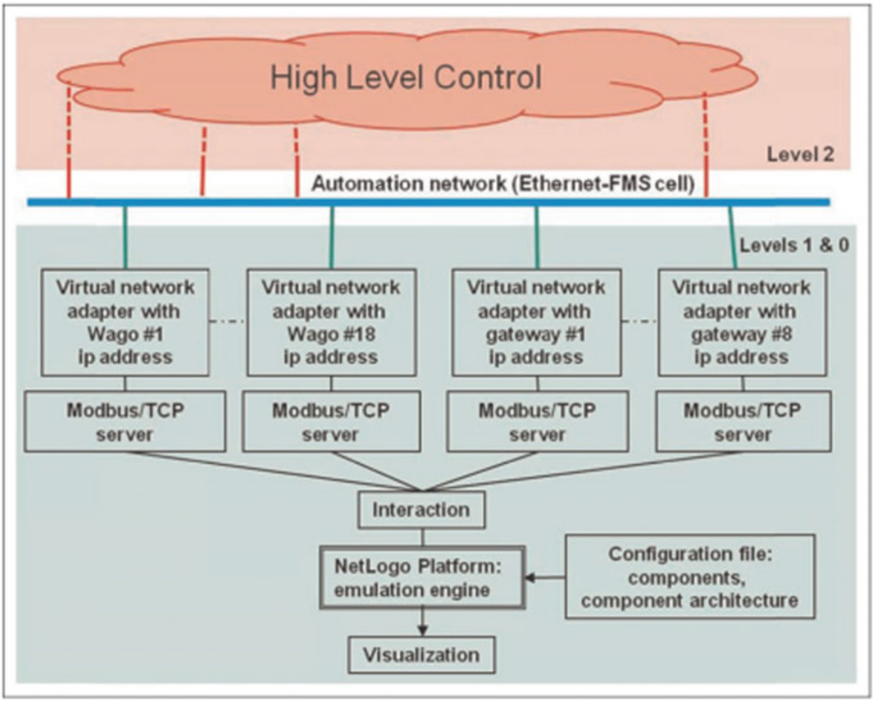


Fig. 5 Arezzo-FMS global scheme [5]

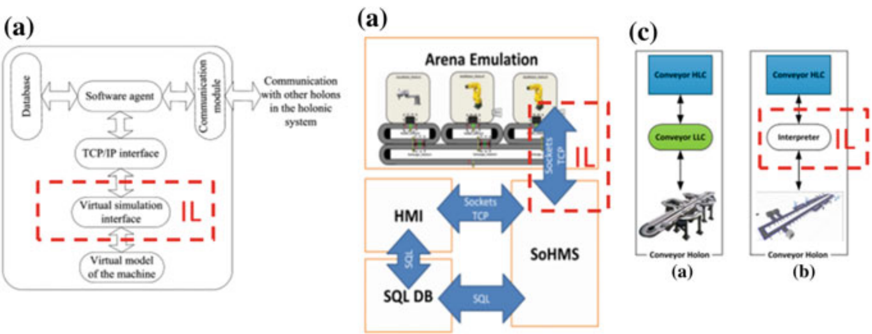


Fig. 6 Some interface layers in literature [7, 11, 12]

3.2.2 Answers to #P2: Performance Indicators

The question of the performance indicators (KPI) is dealt with in two main ways:

1. The emulation model is based on a discrete-event simulation tool, whose outputs are used as KPI, like in [8] for example.
2. The control system has its own KPI output module, which is used in an emulation study in the same way it would be in real cases, like in [7] for example.

However, most of the studies found in literature do not mention the way the KPI are gathered and calculated.

3.2.3 Answers to #P3: Scenario Integration

As far as the authors know about, this question was not deeply treated in a generic way in literature: all the developments were made ad hoc for unique performance evaluations. The only reference to such element can be found in [4] where the scenario is meant to be integrated in parallel with the initial data set of the control environment, but no indication is given on how to achieve this integration in a dynamic environment.

This corresponds to the lack of predefined benchmark exhibited earlier in this article. Now that initiatives such as Bench4Star [4] rose, a generic scenario manager could probably be designed, enabling an easy coupling between Bench4Star and emulation initiatives. This constitutes the purpose of next section, which intends to design a global architecture integrating this scenario manager and specifying the expected characteristics.

4 Proposition of a Generic Implementation Architecture

The scope of this section is to define the general architecture and prerequisites for the most valuable response to the problematic expressed before. Figure 7 introduces the global architecture. It is based on a generic emulation-based architecture (left side of the figure) extracted from the previous analysis of literature. Considering all the works published, the following elements can be defined:

- *Emulation model*: simulation-based dynamic model of the real system.
- *Control system*: Holonic based control system to be evaluated. The human-machine interface was not represented apart here, however it could be.
- *Production database*: for orders management and relationship with tools such as ERP for example.
- *Interface layer*: as previously discussed, this layer intends to ease the switch between emulation-based evaluation and control of the real system.

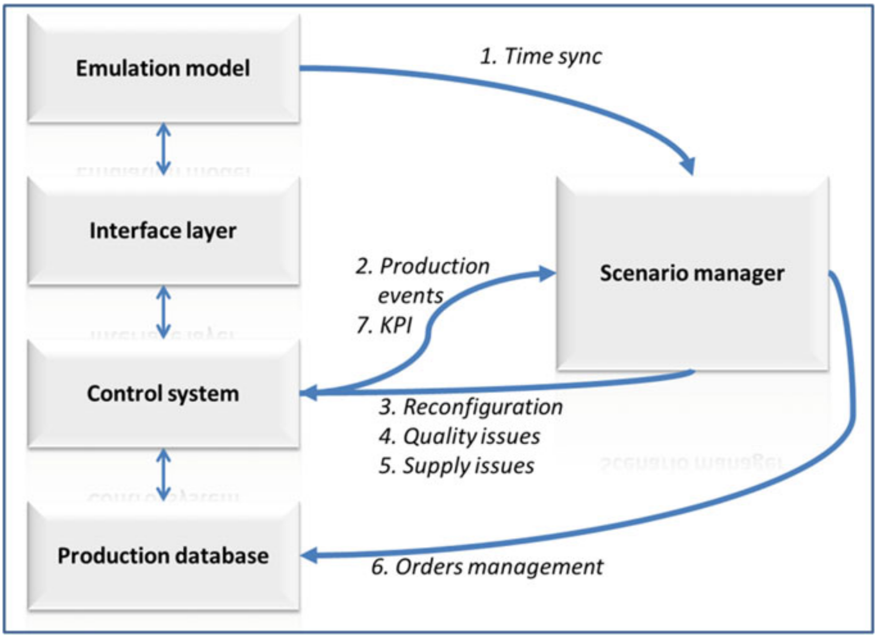


Fig. 7 Integration of a scenario manager in an emulation based control architecture

This last element is the main answer of the architecture to #P1 concerning the integration between holons and emulation. This interface layer shall be standardized in order to be implemented easier. The evolution of control systems, slowly migrating to the cloud [13], makes the problem of interoperability more and more important, and goes thus in the right direction for this purpose. The proposition of using web services-like architectures, designing Service-oriented HMS [15], is probably a first step towards this objective.

Considering #P2 and the problem of performance indicators, none of the options discussed in literature show in our point of view a good adequacy with the objectives of a generic approach for performance evaluation.

The use of the emulation model simulation outputs to calculate the KPI is very interesting for the utilization of machines for example, but seems simplistic from a general point of view, as it prevents from getting KPI about the order management system, or about the behaviours of holons (decision making delays). Furthermore, one of our objectives for the emulation model is to be as lean as possible, so that it can be used in all scenarios without model modification. This is totally impossible with the use of the model for KPI calculation.

Another direction is to design the control system to be able to compute its own KPI. This is a very efficient solution, as this element of the architecture is aware of all the events that can perturb the performances of the overall system. However, it does not seem relevant to modify the design of the control system for emulation purposes: it would be better to use the full control system without modifications. Moreover, the variety of the studied scenarios and the expected associated KPI makes it a huge patch to integrate in the software that might modify the behaviour of the control system.

Therefore, the proposition here is to gather data (label 2. of Fig. 7) or direct KPI (label 7) from the control system and externalize the calculation of the KPI in another element of the architecture. As the expected KPI vary between each tested scenario, this element needs to know about the running scenario and about the actual time of the system. Indeed, the time of the system is dictated by the emulation model, whereas the control system does not necessarily know about it. This element therefore also needs a connection to the emulation model for data gathering (label 1).

This last proposition leads to the definition of a "Scenario Manager", able to modify the behaviour of the control system according to the chosen scenario (labels 3, 4, 5 and 6). For these last features, the scenario manager needs to have an access to the control system in various forms, but all these interactions are probably meant to be at least created for the human-machine interaction. The only one might be quality issues, where the actual information comes from the shop floor in real time execution. In that case, it is the scenario manager that needs to endorse this role and handle most of the random data distributions.

Figure 8 shows a sequence diagram expressing the behaviour of the scenario manager in the case of scenario #PS9 extracted from [4]. This scenario needs a reconfiguration of the system due to machine breakdown. The problem is that the date of the breakdown is determined dynamically considering the departure date of the first shuttle from this machine. Therefore, the scenario manager needs production events to know when it needs to reconfigure the control system to take into account the breakdown of M2.

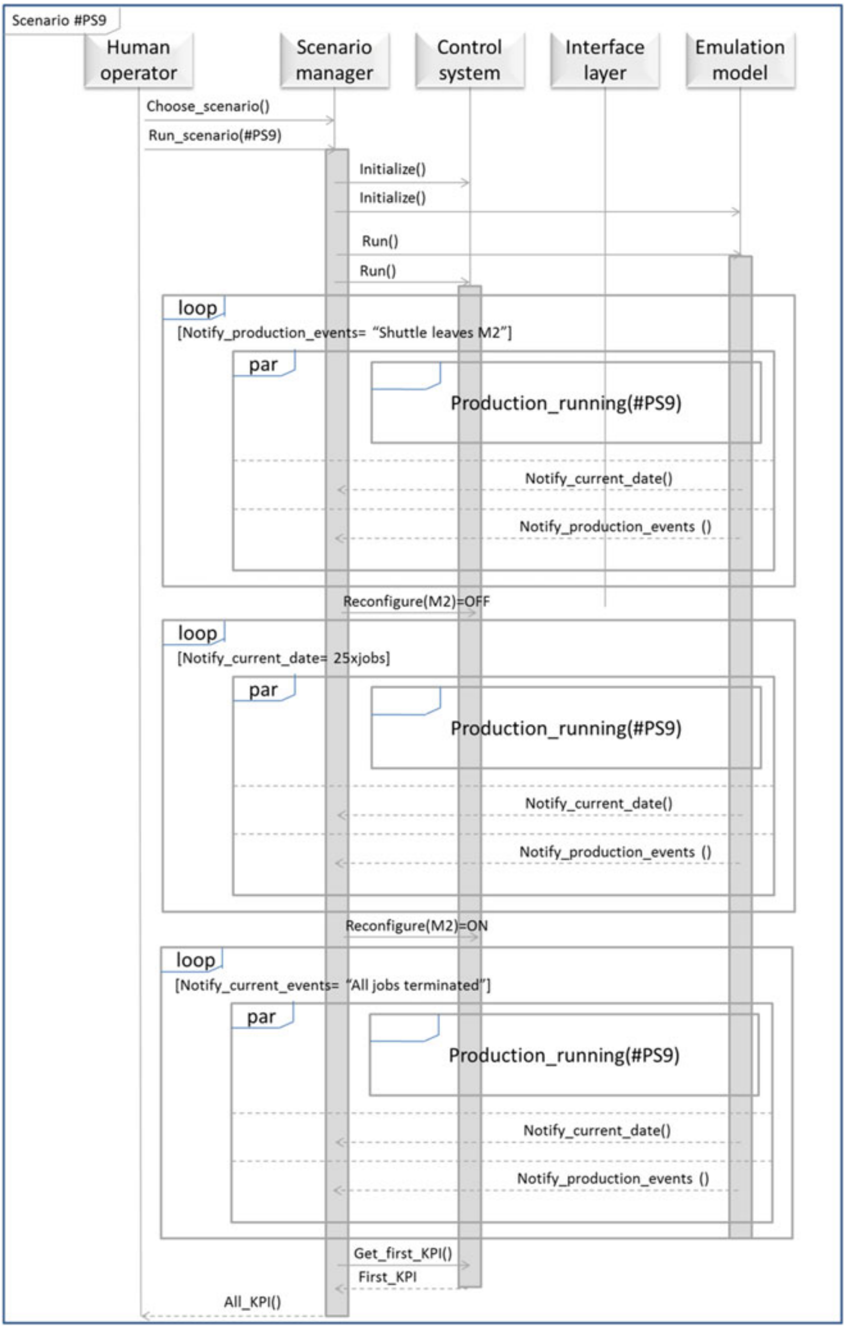


Fig. 8 Sequence diagram of #PS9 scenario integration

5 Conclusion

The objective of this paper was first to provide an extensive overview of the benchmarking initiatives oriented towards the performance evaluation of Holonic Manufacturing Systems. Then, a comparison between all the isolated emulation developments for benchmarking was made, and common features and main problems were exhibited.

Finally, a global architecture dedicated to a generic performance evaluation platform design was suggested. This architecture integrates a scenario manager, whose main specificities were detailed and justified. These features are meant to both integrate the best practices encountered in literature and fulfil the missing aspects to respond to the problems formulated. Basically, the idea is to develop a piece of software integrating a priori all the scenarios of literature benchmarks, with standardized interfaces and which would be able to modify in real time the behaviour of the system (triggering breakdowns, order management, etc.) and generate adequate performance indicators at the end of the scenarios runs. We believe this is the elementary brick missing to be really efficient in performance evaluation, but also a very difficult brick to develop in a generic way.

The main objective now is to foster the globalization of these considerations among the main actors of the domain in order to try and develop a scenario manager able to connect to most of the control systems developed in parallel in the community.

References

1. Cardin, O., Ounnar, F., Thomas, A., Trentesaux, D.: Future industrial systems: best practices of the intelligent manufacturing and services systems (IMS2) French Research Group. IEEE Trans. Ind. Inform. **13**, 704–713 (2017)
2. Indriago, C., Cardin, O., Bellenguez-Morineau, O., Rakoto, N., Castagna, P., Chacòn, E.: Performance evaluation of holonic control of a switch arrival system. Concurr. Eng. https://doi.org/10.1177/1063293X16643568
3. Taillard, E.: Benchmarks for basic scheduling problems, European. J. Oper. Res. **64**, 278–285 (1993)
4. Trentesaux, D., Pach, C., Bekrar, A., Sallez, Y., Berger, T., Bonte, T., Leitão, P., Barbosa, J.: Benchmarking flexible job-shop scheduling and control systems. Control Eng. Practice **21**, 1204–1225 (2013)
5. Berger, T., Deneux, D., Bonte, T., Cocquebert, E., Trentesaux, D.: Arezzo-flexible manufacturing system: a generic flexible manufacturing system shop floor emulator approach for high-level control virtual commissioning. Concurr. Eng. **23**, 333–342 (2015)
6. Cavalieri, S., Terzi, S., Macchi, M.: A Benchmarking Service for the evaluation and comparison of scheduling techniques. Comput. Ind. **58**, 656–666 (2007)
7. Jovanović, M., Zupan, S., Starbek, M., Prebil, I.: Virtual approach to holonic control of the tyre-manufacturing system. J. Manuf. Syst. **33**, 116–128 (2014)
8. Bal, M., Hashemipour, M.: Virtual factory approach for implementation of holonic control in industrial applications: A case study in die-casting industry. Robot. Comput. Integr. Manuf. **25**, 570–581 (2009)

9. Pannequin, R., Morel, G., Thomas, A.: The performance of product-driven manufacturing control: an emulation-based benchmarking study. Comput. Ind. **60**, 195–203 (2009)
10. Mönch, L.: Simulation-based benchmarking of production control schemes for complex manufacturing systems. Control Eng. Practice **15**, 1381–1393 (2007)
11. Gamboa, Q.F., Cardin, O., L'Anton, A., Castagna, P.: Virtual commissioning-based development and implementation of a service-oriented holonic control for retrofit manufacturing systems. In: Borangiu, T., Trentesaux, D., Thomas, A., McFarlane, D. (eds.) Service Orientation in Holonic Multi-Agent Manufacturing, pp. 233–242. Springer International Publishing (2016)
12. Kruger, K., Basson, A.: Validation of a holonic controller for a modular conveyor system using an object-oriented simulation framework. In: Borangiu, T., Trentesaux, D., Thomas, A., Leitao, P., Barata, J. (eds.) Service Orientation in Holonic and Multi-agent Manufacturing, pp. 427–435. Springer International Publishing, (2017)
13. Dobrescu, R., Merezeanu, D.: Simulation platform for virtual manufacturing systems. In: Borangiu, T., Trentesaux, D., Thomas, A., Leitao, P., Barata, J. (eds.) Service Orientation in Holonic and Multi-agent Manufacturing, pp. 395–404 (2017)
14. Rocha, A.D., Barroca, P., Maso, G.D., Oliveira, J.B.: Environment to simulate distributed agent based manufacturing systems. In: Borangiu, T., Trentesaux, D., Thomas, A., Leitao, P., Barata, J. (eds.) Service Orientation in Holonic and Multi-agent Manufacturing, pp. 405–416 (2017)
15. Gamboa Quintanilla, F., Cardin, O., L'Anton, A., Castagna, P.: A modeling framework for manufacturing services in Service-oriented Holonic Manufacturing Systems. Eng. Appl. Artif. Intell. **55**, 26–36 (2016)

Using High Level Architecture in the SEE Project for Industrial Context

Simon Gorecki, Gregory Zacharewicz and Nicolas Perry

Abstract Nowadays, systems are becoming more and more complex. Therefore, modelling & simulation (M&S) are also growing in complexity. That's why it is important to form, train and graduate students and faculty. The National Aeronautics and Space Administration (NASA) host an international event of M&S oriented in solving interoperability problems, Distributed Simulation (DS), and student cooperation. The goal is to use a DS standard: High Level Architecture (HLA) to simulate a moon base. Each team builds a module of it and have to communicate with each other. This context will be propitious for student to learn HLA programming, in order to reuse it in an industrial manufacturing context. The use of HLA is interesting in order to couple already existent heterogeneous works.

Keywords High Level Architecture · Mobile factory

1 Introduction

Nowadays modelling & simulation (M&S) become increasingly important. They allow representing any complex system, to study their functioning and their interactions with the environment. The simulation of a system will allow us to virtually design our subject to anticipate and avoid problems. With the growing technologies, the systems' complexity is increasing and it becomes more and more complicate to simulate them. This caused distributed simulation (DS) to appear: one simulation divides multiple sub-functions that are executed on separated computers.

S. Gorecki (✉) · G. Zacharewicz
IMS UMR CNRS 5218, University of Bordeaux, Bordeaux, France
e-mail: simon.gorecki@u-bordeaux.fr

G. Zacharewicz
e-mail: gregory.zacharewicz@u-bordeaux.fr

N. Perry
Arts et Métiers, Paris Tech ENSAM of Bordeaux, Bordeaux, France
e-mail: nicolas.perry@ensam.eu

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_21

This solution allows solving complexity problems, but it also raises interoperability problems. There is a growing need to educate, train, and certify students, researchers, and faculty for M&S, DS and interoperability problems [1, 2]. Zack Crues, a National Aeronautics and Space Administration (NASA) engineer has suggested to the Simulation Interoperability Standards Organization (SISO) Space Community Forum, to launch an inter-university event which would learn students about the importance of simulation interoperability and simulation: the SEE Project.

This project will serve as a springboard to study the HLA mechanism in order to be reused in industrial domain. A French company has launched an innovative project for setting up a solar power plant. This project deals with different domains including risks. University of Bordeaux supports research in M&S and specifically in DS (Distributed Simulation). Most of these research works have created specific domain simulators. Each of these autonomous simulations is capable of representing a fragment of the global project. One of the last phases of this project is to assemble all these simulations to obtain a global simulation of the problem [3]. However, all these simulations use different technologies and manipulate heterogeneous data, which complicates the assembly. To solve these problems, we will use the HLA standard learned in the SEE project.

2 Distributed Simulation and High Level Architecture

In the computer simulation domain, distributed simulations are among the most useful and powerful applications. Indeed, they consist of several components (often associated with one or more functions) that can be differently processed. All these components are part of a single execution which can be relocated to a different computer/server, hence the term “distributed”. This concept of functions relocation makes possible the loads distribute on different machines and thus increases the efficiency of a program.

One of the advantages of distributed simulation is solving some interoperability problems. Interoperability is the ability of interactions between systems. This problem appears when several systems highly dissimilar (by their internal structure, exchanged data format or semantic data) must communicate. The interoperability problem must be considered if interactions are at data level, service level or process level [4]. These problems involve at least two entities which try to communicate. Consequently, establishing interoperability means to relate two systems together and remove incompatibilities. Incompatibility is the fundamental concept of interoperability. The concept of ‘incompatibility’ has a broader sense and is not limited only to the ‘technical’ aspect as usually considered in software engineering, but also to the ‘information’ and ‘organisation’ ones [4].

Indeed, in distributed simulations the components are modular, they can have a heterogeneous architecture and exchange different structured messages; this allows solving interoperability problems.

In our application case, the notion of distributed simulation will be used with the High Level Architecture standard (HLA) [5]. It is a specification of software architecture, and defines a framework and allows creating global execution consisting of distributed simulations. This framework defines how to create a “global” simulation consisting of several distributed simulations that can communicate with one another. It was originally created by the Office of Defense Modeling and Simulation (DMSO) of US Department of Defense (DoD) to facilitate the assembly of stand-alone simulations with a different architecture. The original goal was the reuse and interoperability of military applications, simulations and sensor. This standard is designed to resolve interoperability and reusability issues between software components.

HLA is an architecture that enables several simulation systems to work together. This is called interoperability, a term that covers more than just sending and receiving data. The systems need to work together in such a way that they can achieve an overarching goal by exchanging services. Interoperability refers to the ability of interactions between systems [6]. It extends beyond the boundaries of any single system, and involves at least two entities. Consequently, establishing interoperability means to relate two systems together and remove incompatibilities.

Another interesting aspect of this technology is synchronization. It enables to dynamically manage interoperability issues. Simulations exchange messages; it must be ensured that messages are sent at the right time, in the right order, and that they do not violate causal constraints. To do this, various systems for synchronization of processes and time management are proposed by HLA.

According to the HLA standard, each simulation participating to the application is called “federate”. A federate interacts with other federates there are forming a group named HLA federation. All of these entities can communicate with each other through a Run-Time Infrastructure (RTI). It is the RTI that will manage the federation, authorize federates to communicate or not, and provide various services such as time management, file or data exchange, etc.

3 Agent Based Approach

Agent-based simulation is one of the most popular approaches for simulating systems. It is based on studding actions and interactions between autonomous components (agents). High Level Architecture (HLA) is based on Distributed Simulation, which is not dedicated to agent model. However, there are already in the past some approaches to use HLA in order to developed agent-based model [7].

4 Run-Time Infrastructure (RTI)

A federation is composed of a set of federates and a Run-Time Infrastructure (RTI) [2]. This RTI provides to federate all functionalities that are described by the specification. Federates can only interact through the RTI. By using HLA, all federates are stand-alone simulations. This allows them to keep their own architecture which can be totally different to the others federates. This principle enables to solve interoperability problems. By keeping their own architecture and using a pivot-language, federates can be plug-in and plug-out to a federation that enables reusability. Federates can "Publish" to inform RTI and other federates about an intention to send information. They also can "Subscribe" to reflect some information created and updated by other federates. This is the basics communication mechanic. The data flow exchanged between all federates is represented in the same form of classical object-oriented programming. There are two types of objects which are exchanged in the HLA standard: Object Class and Interaction Class.

Object Class are time persistent during the simulation. They have attributes that can be updated. For example, in a simulation an Object Class would be a car and its attribute would be a position, a speed or a name. Interactions are not persistent over time and can have parameters. For example, "Start car", "Stop car", "Accident" would be Interaction examples.

These two kinds of objects are described by a XML file named Federation Object Model (FOM) attached to a federate. That's an important point: the FOM describes all the information that will be exchanged between federates, being the only item that will be shared between simulations. This fact has an impact on safety on this technology. Federates are totally autonomous; the only exchanges between them is described by the XML file.

Each federate are single applications which can be executed separately. Originally, HLA was created for reusability; all the components can be executed separately or together with others through a federation. Overall, HAL and the RTI provide information transport, federate time synchronization, and do not violate causal rules.

5 SEE Project

The Simulation Exploration Experience (SEE) is a project initiated by Zack Crues, a National Aeronautics and Space Administration (NASA) engineer. He suggested that the Simulation Interoperability Standards Organization (SISO) Space Community Forum launches an inter-university event which would learn to students about the importance of simulation interoperability by doing it, in order that students become more employable and job-ready than the average college graduates. Called 'Smackdown' [1], this project is supported by industry, academia and government. The first event took place in 2011 and since, it is renewed every year.

In 2017, about 20 teams around the world participated in the project, like Massachusetts Institute of Technology (MIT), Florida Institute of Technology (FIT), Carolina State Colleges (interns at Johnson Space Center), University of Bordeaux (France), University of Genoa (Italy), University of Calabria (Italy), and many others.

The SEE Project met industries, students, teachers and professional associations in a modelling and simulation (M&S) challenge. The goal was to run simulation of a virtual moon base where each team used the last release of HLA (1516e) and industry software in order to create his own module. By using the HLA standard, NASA provided a RTI and several federates that simulate a lunar environment. The RTI and all the federates form a lunar base federation. Each team has worked for months with the technical team, tutorials, and also together through a forum in order to create or upgrade his own federate.

MÄK and Pitch Technologies provided to students evolved RTI software and its associated tutorials [8]. NASA provided also software such as "HLA Starter kit" which simplifies the HLA development [2]. It allowed teams to be more focused on data exchange between their federate by reducing the development complexity. NASA provided also Distributed Observer Network (DON3) that uses a video game graphical engine in order to simulate the scenario in a 3D environment. This one is a special federate which permits to associate a 3D object model to each federate. By sharing through FOM x, y and z coordinates, 3D models could be moved in a virtual space. As the event was international, it was necessary to use remote connection to associate federates with RTI and VPN also provided by NASA.

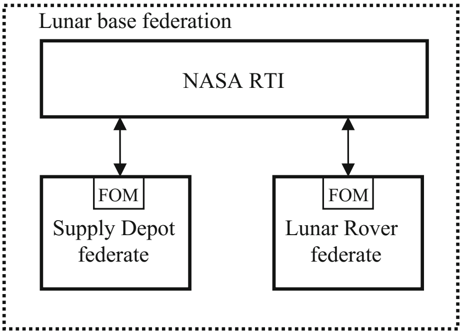
Using all this different tools, the 2017 SEE Project enabled student from Liverpool to create and simulate an Asteroid harvest fleet using mining resources. This team also improved his lunar orbit asteroid. University of Munich built a base which housed observatories and can serve as a means of exploration for the needs of lunar resources. The University of Calabria and Genoa created the Moon City Center (MCC) with interconnected modules provided by Orange Technical College, FACES, Sophia, Liverpool University as astronaut habitat, a standalone residence designed for long term comfort, a greenhouse to supply fresh vegetables for astronauts, a spacecraft Launchpad ready to launch a rocket to the asteroid belt. And finally, the Florida Institute of Technology (FIT) created an exploration rover linked to the Bordeaux's Supply Depot. Both of them can communicate in two ways described in next section.

6 Application

As Bordeaux team, we worked on the Supply Depot Federate and particular on the communication with the FIT's rover as described in Fig. 1.

Our objective was to communicate by using two type of Object that use for communicate with HLA: Object Class and Interaction Class. For that, we had to

Fig. 1 SEE project federation



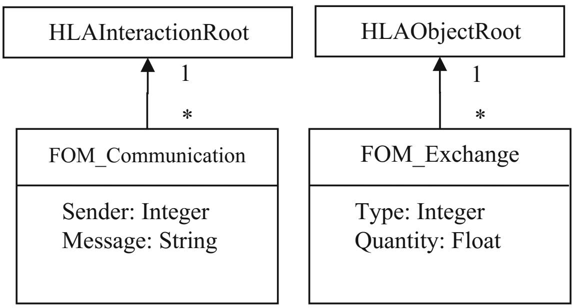
find two types of exchanges: one that need a persistent class to be used with an Object Class, and another with ephemeral notion, used with an Interaction Class.

The ephemeral interaction (mechanism which uses Interaction Class) is the radio communication between the two federates. Each message instantiates an Interaction Class at its reception. After message reception, the federate will save and treat it. The class will destroy itself. For the persistent interaction (mechanism which uses Object Class) the power exchange from the Supply Depot to the rover was considered. This class is created at federate initialization, and when a power exchange is needed the two federates will use this class for energy exchange. In our process algorithm, to trigger a power exchange (when the rover is low battery), the rover has to be near the supply depot and send to it an energy request message.

The Federation Object Model has the role of describing two types of objects, described below. This specification is necessary for ensuring communication in HLA environment.

We can see in the above UML Diagram (Fig. 2) the lunar supply depot Federation Object model representation. As an object-oriented programming language, the "communication" class (left) inherits from the Interaction Class (HLAInteractionRoot), and the "power exchange" class (right) inherits from the Object Class (HLAObjectRoot). The parameters and attributes are specified below the class name. The diagram represents the description of the pivot format that enables HLA interoperability. Indeed, in our case, we are trying to establishing communication

Fig. 2 Lunar supply depot federation object model (FOM)



between two simulations (supply depot and rover) which are totally different. They haven't got the same structure but, by using the FOM technology, they will be able to communicate through HLA.

From a technical point of view, federates are developed in Java on Eclipse IDE. HLA concepts are implemented in several frameworks. The HLA library provided by Pitch allows implementing HLA functionalities. And the NASA HLA Starter kit is an overlay to simplify the HLA development.

For this project, the Bordeaux team had to build a 3D object model in order to be represented by the graphical engine (Distributed Observer Network). For this work, we used the Industrial design software SolidWorks with an additional plugin to export it in geometry definition file format (.obj).

We can see on the picture in Fig. 3 a simulation screenshot that takes places in a past event with a supply depot having interaction with a rover. The visual render was made with DON3, a graphical engine developed by NASA.

Figure 4 shows the UML communication diagram sequence between the two federates, and the RTI. Supply Depot and Rover are designed for communication. The Bordeaux team and FIT team had to work together to establish the scenario below.

In our case, the communication by messages allows to exchange text messages. These messages aren't persistent over the time, so they will be set as Interaction class. Once the message received, the object will be destroyed. Its main function during the simulation is to test whether federates are operational. Contrary to the other communication mean, transfer energy between the depot and the rover will be persistent over the simulation; and therefore it will be defined as an Object class.

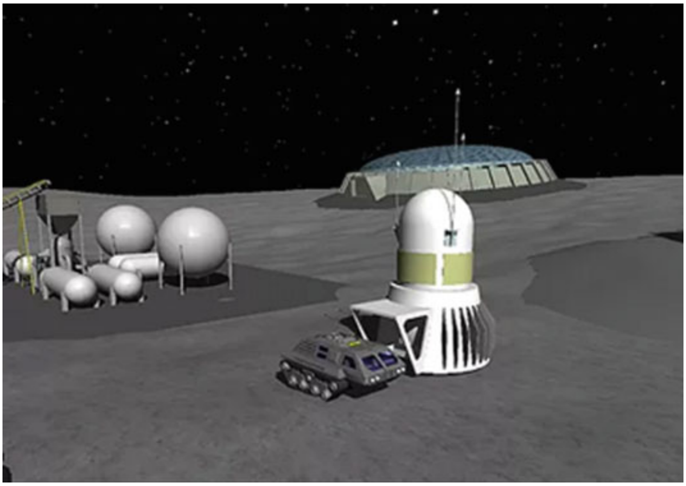


Fig. 3 SEE project simulation screenshot

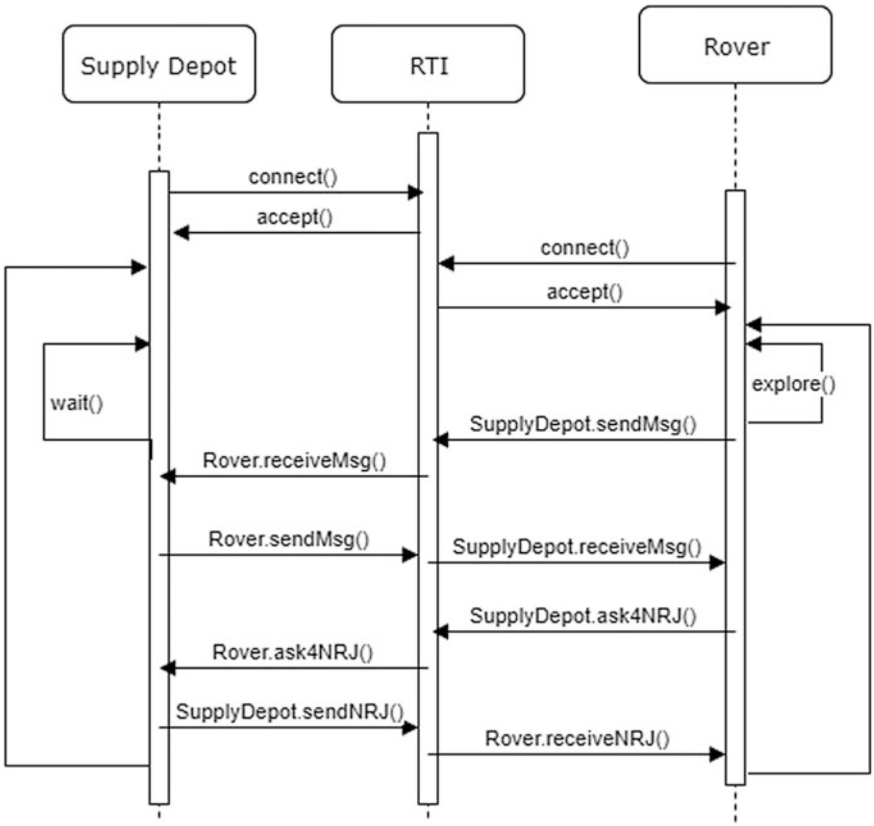


Fig. 4 UML communication diagram sequence

7 Industrial Application

Once the SEE project completed, we were able to apply HLA knowledge in an industrial context. This context is a French-Morocco company wanting to design, develop and control a solar power plant. This project consists in installing solar panels fields in several countries in order to provide electricity in areas which are not powered so far. In terms of implementation and difficulties caused by the project complexity, this innovation gave rise to several simulations that solved several problems:

- Simulation for the designing of solar transmitter supporting structure.
- Simulation for defining the structure foundations of the solar panels field, depending on the ground structure.
- Study and dimensioning of the mobile factory size, cost, etc., based on the demand

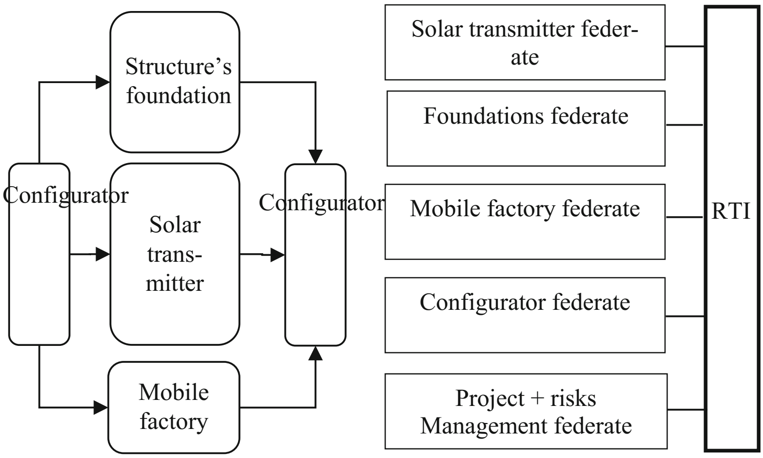


Fig. 5 Global simulation as BPMN and HLA view

- Study of project management integrating risks: evaluating risks probabilities in project management

However, these simulations where made by several persons, not in the same time, and with different languages. So, they aren't compatible with each other. That's why we will use the HLA in order to solve interoperability issues and assemble them in order to create a global simulation composed of each one of those simulations.

We can see on the left side in Fig. 5 a representation of all needed process workflow simulations. The three main simulations take and store data on a configurator which acts as database and GUI. The risks management project is not considered in this workflow because it acts separately. On the right side in the figure one can see what could be the project architecture by using HLA technology. In this configuration, each simulation could be a federate connected to the RTI. The configurator federate acts as data centre of the federation, and the control centre drives scenarios.

8 Conclusion

The works developed in the frame of 2017 SEE Project has been a very instructive adventure at many levels for students. It has permitted to develop distributed simulation components. Concerning the social aspect, it has allowed exposing oneself to problems of interoperability through the simulation of lunar modules. It was also a very good exercise to understand and master the simulations distributed through the HLA specification. It made possible running widely used technologies such as Virtual Private Network (VPN) in order to communicate between each federated and the RTI hosted on NASA servers, or industrial design software like SolidWorks.

The SEE project allowed us to study and understand the HLA mechanisms in order to apply them in an industrial context. The state of-the-art revealed that distributed simulation and the HLA standard can provide an interesting answer to couple these heterogeneous works. The future step will consist in defining, according to HLA Federation Development Process (FEDEP), the behaviour of each federate to assemble simulations.

Acknowledgements From a general point of view, this project brings much experience at the human level. Indeed, it has allowed us to work with major universities around the world. For that we would like to thank you NASA's engineers for their support and the developers of the SEE Project providing all participants with a memorable, interactive, problem-solving experience which can contribute importantly to the workforce of the future.

References

1. Elfrey, P.R., Zacharewicz, G., Ni, M.: SMACKDOWN: adventures in simulation standards and interoperability. In: Proceedings of Winter Simulation Conference, pp. 3963–3967 (2011)
2. Falcone, A., Garro, A., Anagnostou, A., Chaudhry, N.R., Salah, O., Taylor, S.J.E.: Easing the Development of HLA Federates: The HLA Development Kit and Its Exploitation in the SEE Project, Distributed Simulation and Real Time Applications (DS-RT), pp. 50–57 (2015)
3. Gorecki, S., Zacharewicz, G., Perry, N.: Using High Level Architecture to Combine Simulations in Company Context Mobile Factory. International Multidisciplinary Modeling & Simulation Multiconference, Bordeaux (2017)
4. Zacharewicz, G., Labarthe, O., Chen, D., Vallespir, B.: Multi Agent/HLA Enterprise Interoperability (Short-Lived Ontology Based), The International Workshop on Modelling and Applied Simulation, Puerto de la Cruz, Spain, pp. 187–196 (2009)
5. IEEE Computer Society: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA). Federate Interface Specification. IEEE Computer Society, USA, Available from standards.ieee.org (2010). Last accessed 1 June 2017
6. Zacharewicz, G., Chen, D., Vallespir, B.: Short-lived ontology approach for agent/HLA federated enterprise interoperability. Interoperab. Enterprise Softw. Appl. 329–335 (2009)
7. Lees, M., Logan, B.: Distributed simulation of agent-based systems with HLA. ACM Trans. Model. Comput. Simulat. TOMACS, 881–888 (2007)
8. Möller, B., Eriksson, S., Wihlborg, A.: Developing an HLA tutorial: philosophy and best practices. In: Proceedings of 2012 Fall Simulation Interoperability Workshop, Simulation Interoperability Standards Organization, 12F-SIW-047 (2012)

Control of Rail-Road PI-Hub: The ORCA Hybrid Control Architecture

Nhat-Vinh Vo, Thierry Berger, Thérèse Bonte and Yves Sallez

Abstract In the innovative field of Physical Internet, the development of efficient PI-cross-docking hubs (denoted PI-hub) allowing quick, efficient and flexible transfer of containers is a cornerstone. The paper addresses the control of such facilities and focuses more particularly on a “rail-road” PI-hub. After a review of the related control issues, an instance of the ORCA hybrid control architecture is proposed. The focus is held on the PI-containers routing in a perturbed environment. More specifically, predictive and reactive routing strategies are presented. The simulation environment, protocol and results are then detailed.

Keywords Physical internet · Logistics · Cross-dock · Hybrid control architecture

1 Introduction

The innovative Physical Internet (PI) paradigm, inspired from the Digital Internet, aims to face the global logistics sustainability grand challenge [1, 2]. By analogy with data packets, the goods are encapsulated in modularly dimensioned easy-to-interlock smart containers, called PI-containers. This last are designed to flow efficiently in connected networks of logistics services. The development of efficient PI-cross-docking hubs (denoted PI-hubs), allowing quick and flexible transfer of PI-containers is mandatory for the PI development [2, 3]. Different modal transfers (e.g. road to train, road to road, ship to train…) must be considered, with potential impact not only economical but also environmental and social [4]. To allow the development of an efficient PI-network, the control architecture of

N.-V. Vo · T. Berger (✉) · T. Bonte · Y. Sallez
University of Valenciennes and Hainaut-Cambrésis, LAMIH UMR CNRS n°8201, 59313 Valenciennes, France
e-mail: Thierry.Berger@univ-valenciennes.fr

N.-V. Vo
e-mail: NathVinh.Vo@univ-valenciennes.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_22

PI-hubs has to be “globally predictive” to ensure an overall performance (i.e. quick and efficient transfer of PI-containers), and “locally reactive” to deal with perturbations [3, 5].

In this context, the paper is related to the ANR project PI-NUTS (Physical Inter-Net cross-docking hUb conTrol System) [6], and proposes a hybrid control architecture to manage a rail-road PI-hub.

This paper is organized as follows. First, the functioning of a rail-road PI-Hub and inherent control issues are described. Section 3 presents hybrid control architectures in general and our proposal ORCA (Optimized and Reactive Control Architecture) in particular. Applied to the control of a PI-hub, an instantiation of ORCA is detailed in Sect. 4. A simulation study, described in Sect. 5, demonstrates the effectiveness of the proposed architecture. Finally, conclusion and perspectives are offered.

2 Rail-Road PI-Hub Presentation

A *cross-docking hub* is defined as “the process of moving merchandise from the receiving dock to shipping without placing it first into storage locations” [7–9]. In a PI-network, a rail-road cross-docking-hub is an attractive modal solution, since railway allows a reduction of the environmental impacts in freight transportation (e.g., less nuisance emission, usage of renewable energy). The aim of this type of PI-hub is to allow efficient transfers of PI-containers among trains and trucks without dismantling of the trains. In a rail-road PI-hub, several PI-containers flows can be distinguished (rail → road, rail → rail and road → rail) [3]. In the PI-hub, three specific zones are devoted to these flows. The present paper is focused on the “rail → road” zone. Figure 1 describes this zone and highlights two main parts: sorting and manoeuvring areas.

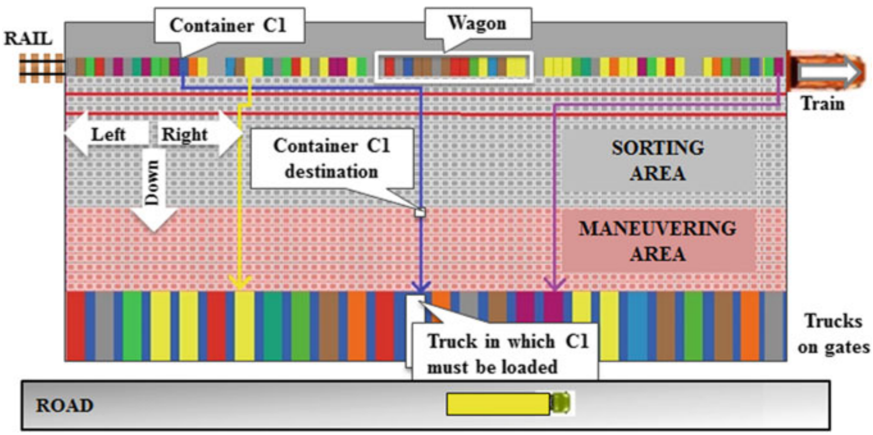


Fig. 1 Sorting and manoeuvring areas in the “rail → road” zone [6]

- Once the PI-containers have been unloaded from a train, they are engaged in a routing process in order to be loaded each in their outbound truck. The aim of the sorting area is to sort the incoming PI-containers and to route them toward the manoeuvring area. The PI-containers are handled (left, right or down) by a grid of PI-conveyors [10].
- The manoeuvring area allows, via a dedicated equipment (e.g. stackers), to turn the PI-containers and to load them in the trucks.

To manage the "rail → road" zone, three main issues must be treated [6]:

- Allocation of the PI-containers to the trucks taking care of their next destination [11].
- Scheduling of trucks on the gates to locate adequately the trucks in front of the different wagons at the right time.
- Routing of incoming PI-containers from the train to the trucks.

All these issues must be treated by the control architecture to obtain a very fast transfer of the PI-containers. Indeed, to insure the efficiency of the PI-network, any delay in the PI-hub must be avoided [1, 3]. In addition, the control of the "rail → road" zone must equally deal with different degrees of variability and uncertainty (e.g., variation of the flow of the incoming PI-containers, trucks in late, failed PI-conveyors). This situation clearly militates to the adoption of a hybrid control architecture that integrates global predictive capabilities and efficient local reactive abilities to face perturbations. This type of control architecture is presented in the next section.

3 Hybrid Control Architectures

After a brief literature review on control architectures, our ORCA solution is presented.

3.1 Literature

According to Trentesaux [12], the control architectures can be grouped in three classes: class I contains typical hierarchical architectures, class III contains heterarchical architectures and class II contains hybrid or semi-heterarchical architectures aiming to gain from benefits of class I and III. In these class II architectures, the hierarchical part (i.e., class I) is responsible for the predictive and global optimization, while the heterarchical part (i.e., class III) allows reactivity and local optimization.

Hybrid control architectures of class II have been implemented in the literature, as for example: ADACOR [13], PROSA [14], POLLUX [15] and ORCA [16].

The next paragraph details the Optimized and Reactive Control Architecture (ORCA), developed in the LAMIH laboratory.

3.2 ORCA

From a generic point of view, ORCA is based on the notion of entities to be controlled. Firstly, the entities can be controlled by a predictive strategy optimizing the global performance of the whole entities. Secondly, each entity can be controlled locally by a reactive strategy optimizing its local performance. At one moment, an entity is either under predictive or reactive control. From an operational viewpoint, an entity can be in one of the two modes: "executing" or "autonomous".

In "executing" mode three major steps occur (see Fig. 2):

1. The predictive strategy (localized at the global control layer) determines the plans (optimizing the global performance) that the entities must follow.
2. The plan to be executed by each concerned entity is given to it.
3. The plan is executed by the entity at the local control layer.

In "autonomous" mode (at local control layer) two steps occur (see Fig. 2):

1. The entity defines autonomously its own plan (optimizing its local performance).
2. The plan is executed by the entity.

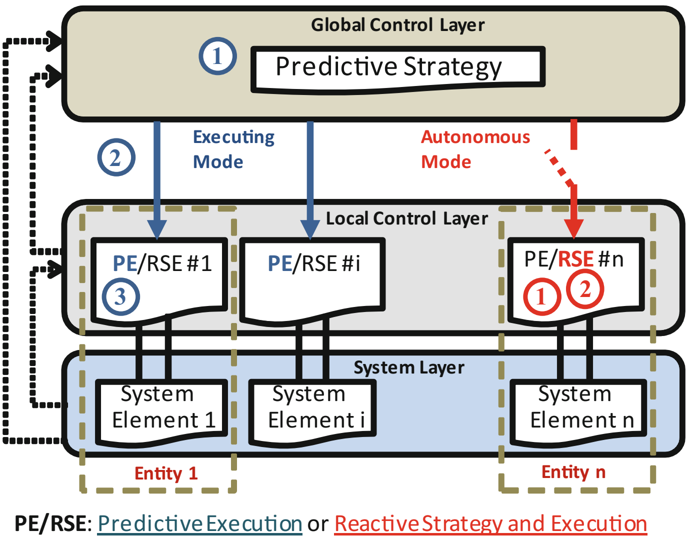
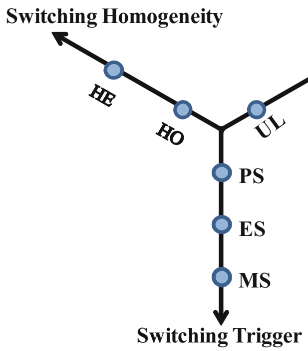


Fig. 2 Two operating modes in ORCA



Axis	Legend
Switching Trigger	**PS** : Periodical Switching
	ES : Event based Switching
	MS : Mixed Switching
Switching Homogeneity	**HO** : HOmogeneous switching
	HE : HEterogeneous switching
Switching Position	**UL** : Switching located at Upper Layer
	LL : Switching located at Lower Layer

Fig. 3 Characteristics of the switching mechanism

Initially, all the entities are in “executing” mode. If a perturbation occurs a switching mechanism can change the mode of the entity in “autonomous”. Once the perturbation is recovered, the switching mechanism can disable the “autonomous” mode and enable the “executing” mode. For the switching mechanism several characteristics are considered based on Fig. 3:

- ***Switching trigger***: the process leading to switching decision can be implemented according to the three following modalities:
  - *Periodical Switching* (PS): the process is integrated in the local control layer of the entity and trigged periodically.
  - *Event based Switching* (ES): the process is trigged once an external event influences the entity.
  - *Mixed Switching* (MS): the process is trigged thanks to the periodical modality and the external event.
- ***Switching position***: The switching can be located on one of the two layers:
  - The upper layer (UL—global control layer).
  - The lower layer (LL—local control layer).
- ***Switching homogeneity***: the switching homogeneity corresponds to the modality that entities are given autonomy:
  - The heterogeneous switching (HE) for only concerned entities.
  - The homogeneous switching (HO) for all entities.

In the next section, the generic concepts introduced in ORCA are instantiated in the PI-Hub context.

4 ORCA for-PI-Hub

In the PI-NUTS project, the scheduling and allocation problems have been studied in [11] and are solved before the arrival of the train. In this paper, only the routing issue is considered (i.e. all the trucks are assumed efficiently located at the gates and a destination gate is given for each incoming PI-container).

As depicted in Fig. 4, the predictive and reactive routing optimization strategies are respectively supported by the global control layer and the local control layer of the architecture. In the global control layer, a predictive routing method generates a routing path (i.e. ordered list of the PI-conveyors units to use) for each PI-container.

Initially, in the local control layer, each PI-container is in the “executing” mode and follows its routing path (provided by the global control layer). According to the switching principle, when a perturbation occurs on its path (i.e. failure of PI-conveyors), the “autonomous” mode is activated. So the affected PI-container applies a reactive routing strategy and tries to find a solution to return to the planned path. The switching trigger is located in the local control layer of each entity (i.e. the PI-container) and allows transitions between “executing” and “autonomous” modes. According to the previously introduced typology (see Fig. 3), our switching mechanism is of “LL-ES-HE” type.

Predictive Routing Strategy: At the global control layer, the path of each PI-container is planned by a heuristic method, based on the following routing principle:

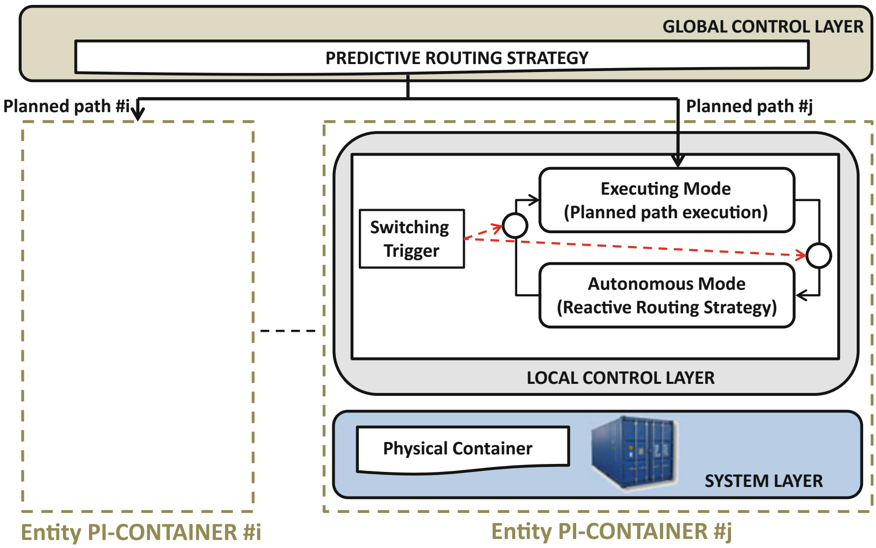


Fig. 4 ORCA PI-Hub illustration

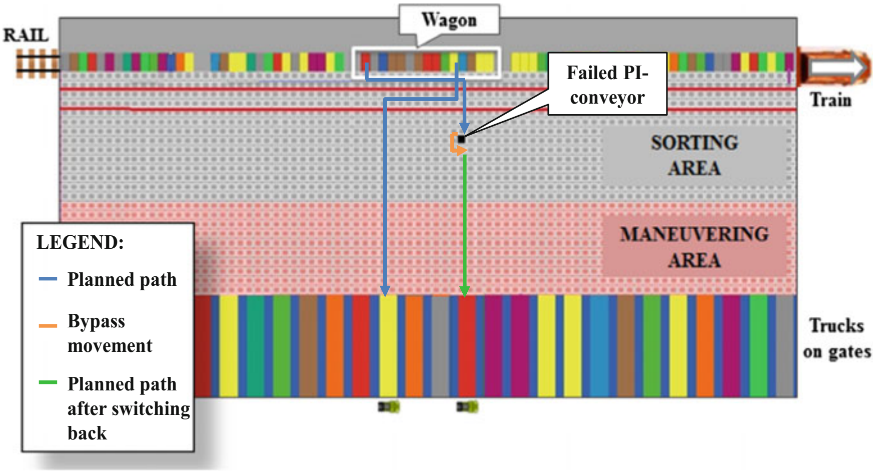


Fig. 5 Illustration of the predictive and reactive routing strategies

- When unloaded from the wagons, the PI-containers are first moved in front of their destination location (see Fig. 1), by the two first lines of PI-conveyors of the sorting area. The first line of PI-conveyors is used for the PI-containers to move from left to right and the second line for the ones that move from right to left (see Fig. 5). To prevent any conflict on these two lines (i.e. two or more PI-containers need the same PI-conveyor at the same time), a predefined rule of priority is applied: priority is given to PI-containers moving down, then to the ones going to the left on the second line, and finally to the ones going to the right on the first line.
- When the PI-containers arrive in front of the destination location, they go down through the sorting area.

Figure 5 illustrates the planned paths of two incoming PI-containers, the first one to the right and the second one to the left.

Reactive Routing Strategy: This strategy, supported by the local control layer of each PI-container, is activated by the switch mechanism in case of perturbation. So the PI-container leaves the planned path and bypasses the failed PI-conveyor by a left or right movement according to the available space. An example of bypass movement is shown in Fig. 5. When the bypass movement is finished, the initial planned path can be resumed. The PI-container joins its previously allocated path, so it is not necessary to redo the global optimization.

The effectiveness of these strategies is investigated in the following section.

5 Simulation Results

In order to validate the effectiveness of the proposed control architecture, a simulation platform has been designed using the multi-agent environment NetLogo [17]. It is well adapted for prototyping of reactive multi-agent systems. It provides three types of elements: patches, turtles and links. PI-conveyors are modelled by patches, while PI-containers, trucks and wagons are modelled by turtles. Links define temporary relationships between PI-containers.

To evaluate the performance of the routing strategies, three policies are investigated:

- The "Policy A" is representative of a functioning without perturbation (i.e. the planned paths are respected for each PI-container).
- In the "Policy B" perturbations occur and PI-containers remain in the executing mode (without switching possibility). PI-containers are blocked when they encounter a failed PI-conveyor. In this case they must be manually evacuated. The manual evacuation is triggered only when all unblocked PI-containers are evacuated from the sorting area.
  To avoid a too important distortion in the comparison predictive versus reactive behaviours, the manual evacuation time is assumed to be not too long (see Table 1).
- In the "Policy C" perturbations occur and the proposed control approach is active (i.e. switch between "executing" and "autonomous" modes).

The simulation data are described in Table 1.

Two studies are performed. The first one concerns the global performance of the PI-hub. The second one evaluates the influence of the PI-containers size in function of the number of failed PI-conveyors. For each study, 300 simulation runs were performed (each simulation run corresponds to the unloading of a train). The three policies were evaluated for each simulation run.

Table 1 Simulation data

Number of wagons in a train	30
Number of wagons in the same group for unloading	5
Time to move from a group to the following one	3 mn
Size of a PI-conveyor	1.2 m × 1.2 m
Possible sizes of PI-containers	1.2; 2.4; 3.6; 4.8; 6; 12 m
Time to go from a PI-conveyor to a neighbouring PI-conveyor in normal operation	12 s
In "manual" operation (Policy B)	24 s
Time to unload a PI-container from a wagon	1 mn
Time to load a PI-container in a truck in normal operation	2 mn
In "manual" operation (Policy B)	4 mn

The retained performance indicator is the evacuation time (i.e. time between un-loading the first PI-container from the train and loading the last PI-container onto its truck).

Concerning the perturbations, several assumptions are taken into account:

- The two first lines of PI-conveyors in the sorting area are considered to be of high reliability without any failure.
- The failures affecting the PI-conveyors are distributed among the areas located in front of the five wagons (i.e. each area has 0, 1 or 2 failed PI-conveyors). The maximum number of failed PI-conveyors is 10.

Study#1: Global performance evaluation

Figure 6 shows the evolution of the PI-hub performance for the three policies. Some interesting results can be noticed:

- The evacuation time obtained with Policy C is lightly higher than with Policy A, showing the good performance of ORCA.
- The performance of Policy C is nearly twice better than Policy B showing the efficiency of the reactive strategy to face perturbations.

The poor performances of Policy B are due to the disorder generated by the failed PI-conveyors. According its size, a blocked PI-container leads to several "unusable" PI-conveyors, and can induce other PI-containers blockage (by a cascading effect). Figure 7 gives an example of a total PI-hub blockage with large PI-containers.

Study#2: Influence of the number of failures

The evacuation times are represented according to the number of failed PI-conveyors, varying from 0 to 10 (0 corresponds to no perturbation).The average evacuation time of each train is exhibited in Fig. 8 (respectively Fig. 9) in case of large size (respectively small size) PI-containers.

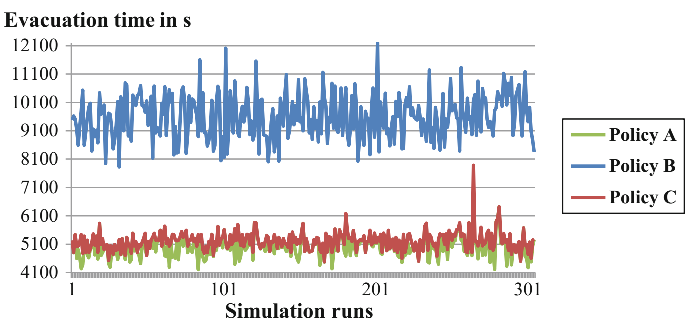


Fig. 6 Evacuation time according to the three policies

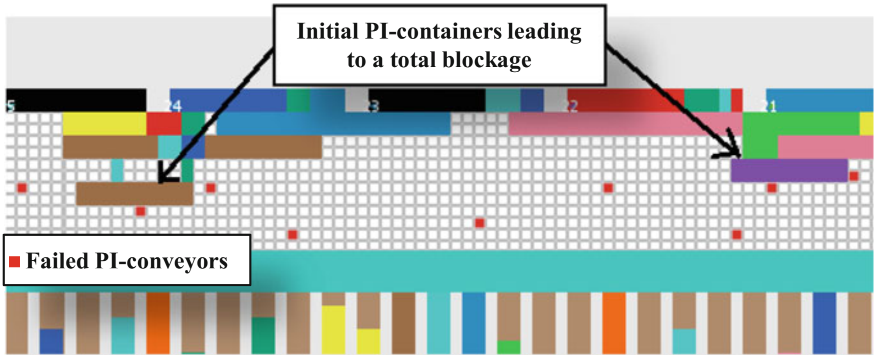


Fig. 7 Example of total blockage with large size PI-containers

Fig. 8 Evacuation time in case of large size PI-containers

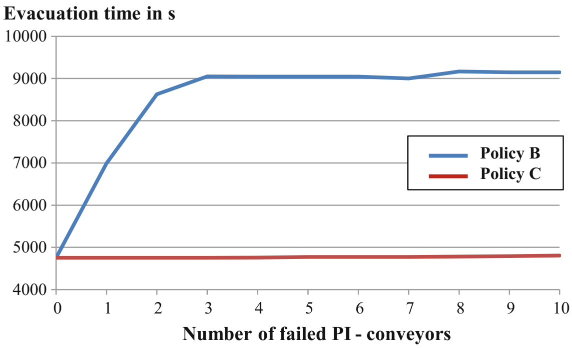
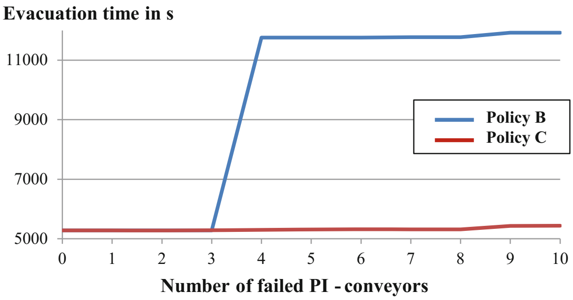


Fig. 9 Evacuation time in case of small size PI-containers



For a train loaded with large PI-containers (i.e. maximum size: 12 m), the difference between Policy B and Policy C is manifest (i.e. the evacuation time for Policy B is 80% higher than for Policy C). As soon as a PI-conveyor fails, the large blocked PI-container quickly leads to a total blockage of the system, as can be observed in Fig. 7).

For a train loaded with small PI-containers (i.e. minimum size: 1.2 m), Policy B and Policy C are fairly close if the number of failed PI-conveyors is low. It is due to the fact that a small PI-container blocked by a failure generates fewer blockages than a large PI-container. In the case of few perturbations, the manual evacuation mode is therefore triggered very late and its influence on the total evacuation time is significantly reduced.

It can be stated that the hybrid control architecture ORCA (notably in case of perturbation) is effective. Moreover, architecture ORCA switches between executing and autonomous modes only when necessary without oscillation: the number of switching decisions for each PI-container cannot be greater than the number of failed PI-conveyors met. The duration of the "autonomous" mode is highly variable depending of several parameters (e.g., number of PI-containers in the vicinity, PI-containers size, number and locations of failed PI-conveyors).

6 Conclusion and Perspectives

In this paper, the general issues concerning the control of the rail-road PI-hub has been introduced and an instantiation of the ORCA hybrid control architecture has been proposed. Via a switching mechanism, this hybrid control architecture can operate in "executing" mode to ensure an overall efficient performance (i.e. quick transfer of PI-containers) and in "autonomous" mode to deal with perturbations. The focus has been set on the routing of the PI-containers in the "rail → road" zone of a PI-hub. The simulation results showed that the proposed ORCA architecture is able to face important local perturbations affecting the conveying network.

In short time, a first prospect is to improve the robustness of the reactive routing strategy by allowing backward movements of the PI-containers to bypass failed PI-conveyors.

In midterm, the next challenge to achieve is to develop efficient strategies for the two others issues (i.e., scheduling of trucks, PI-containers allocation) and to integrate them in ORCA.

Acknowledgements The authors would like to especially thank the French National Research Agency (ANR) which supports this work via the granted PI-NUTS Project (ANR-14-CE27-0015).

References

1. Montreuil, B.: Toward physical internet: meeting the global logistics sustainability grand challenge. Logist. Res. **3**, 71–81 (2011). https://doi.org/10.1017/CBO9781107415324.004
2. Treiblmaier, H., Mirkovski, K., Lowry, P.B.: Conceptualizing the physical internet: literature review, implications and directions for future research. Annu. Eur. Res. Seminar, 1–17 (2016)

3. Ballot, E., Montreuil, B., Thivierge, C.: Functional design of physical internet facilities: a road-rail hub. In: Progress in Material Handling Research, vol. 12, pp. 28–61, MHI, Charlotte, NC (2014)
4. Ballot, E., Gobet, O., Montreuil, B.: Physical Internet Enabled Open Hub Network Design for Distributed Networked Operations. In: Borangiu, T., Thomas, A., Trentesaux, D. (eds.) Service Orientation in Holonic and Multi-Agent Manufacturing Control, vol. 402, pp. 279–292. Springer, Berlin Heidelberg (2012)
5. Sallez, Y., Berger, T., Bonte, T., Trentesaux, D.: Proposition of a hybrid control architecture for the routing in a physical internet cross-docking hub. IFAC-Papers OnLine **48**, 1978–1983 (2015). https://doi.org/10.1016/j.ifacol.2015.06.378
6. PI-NUT: ANR PI-NUTS (2015). http://www.univ-valenciennes.fr/pi-nuts
7. Apte, U., Viswanathan, S.: Effective cross docking for improving distribution efficiencies. Int. J. Logist. Res. Appl. **3**, 291–302 (2000)
8. Van Belle, J., Valckenaers, P., Cattrysse, D.: Cross-docking: state of the art. Omega **40**, 827–846 (2012). https://doi.org/10.1016/j.omega.2012.01.005
9. Pach, C., Sallez, Y., Berger, T., et al.: Routing management in physical internet crossdocking hubs: study of grouping strategies for truck loading. Adv. Prod. Manag. Syst. Innov. Knowled. Based Prod. Manag. Glob. World, 483–490 (2014)
10. MHIA: Glossary of the Material Handling Industry of America. (2011). http://www.mhi.org/glossary Last accessed 30 May 2017
11. Walha, F., Bekrar, A., Chaabane, S., Loukil, T.M.: A rail-road PI-hub allocation problem: Active and reactive approaches. Comput. Ind. **81**, 138–151 (2015). https://doi.org/10.1016/j.compind.2016.04.007
12. Trentesaux, D.: Distributed control of production systems. Eng. Appl. Artif. Intell. **22**, 971–978 (2009). https://doi.org/10.1016/j.engappai.2009.05.001
13. Leitão, P., Restivo, F.: ADACOR: A holonic architecture for agile and adaptive manufacturing control. Comput. Ind. **57**, 121–130 (2006). https://doi.org/10.1016/j.compind.2005.05.005
14. Van Brussel, H., Wyns, J., Valckenaers, P., et al.: Reference architecture for holonic manufacturing systems: PROSA. Comput. Ind. **37**, 255–274 (1998). https://doi.org/10.1016/S0166-3615(98)00102-X
15. Jimenez, J.-F., Bekrar, A., Zambrano-Rey, G., et al.: Pollux: a dynamic hybrid control architecture for flexible job shop systems. Int. J. Prod. Res. 1–19, (2016). https://doi.org/10.1080/00207543.2016.1218087
16. Pach, C., Berger, T., Bonte, T., Trentesaux, D.: ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. Comput. Ind. **65**, 706–720 (2014). https://doi.org/10.1016/j.compind.2014.02.005
17. Wilensky, U.: GasLab - an Extensible Modeling Toolkit for Exploring Micro- and Macro-Views of Gases. In: Roberts, N., Feurzeig, W., Hunter, B. (eds.) Comput. Model. Simul. Sci. Educ., pp. 151–178. Springer, Berlin (1999)

Cyber-Physical Logistics System for Physical Internet

Patrick Pujo and Fouzia Ounnar

Abstract The 4th industrial revolution undeniably impacts digitally logistics systems, which will greatly evolve. Logistics networks will strengthen their strategic and economic roles. With digitization, logistics networks become a completely integrated ecosystem, which includes new ICT advances such as Cyber-Physical System (CPS). This digital transformation led to the concept of Cyber-Physical Logistics System (CPLS). The paper describes the operation of a CPLS for Internet physical deployment based on the holonic paradigm.

Keywords Cyber-Physical logistics system · Analytic network process
Multicriteria decision making · Logistics network · Physical internet

1 Introduction

The new industrial revolution called 'Industry 4.0' will undoubtedly influence the logistics world: we are already talking about 'Logistics 4.0'. This new revolution is based on a strong interconnection of objects and/or actors, and also of the real world and the virtual world via ICT innovations [1]. This digital transformation more generally affects the industrial ecosystem. According to [2], the digital revolution gradually erases the boundaries between BtoB and BtoC, by directly interconnecting offers and demands. The interactive and transparent collaboration offered by the evolution of the Internet world eliminates the traditional organization in sectors, branches and professions, and allows the customer to participate to the creation of added value, within a co-design hybrid (customer-provider) relationship. In this context, the advantage will go to the one who will know, by associating his customers, to create unimaginable opportunities arising from complementary

P. Pujo (✉) · F. Ounnar
Aix-Marseille Univ., CRET-LOG, Aix-en-Provence, France
e-mail: patrick.pujo@univ-amu.fr

F. Ounnar
e-mail: fouzia.ounnar@univ-amu.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_23

business. The technological leaps observed in recent years, linked in particular to the development of the numerical tools, have led to paradigm shifts that had to integrate these new elements.

In the manufacturing world, 'Industry 4.0' is based on various technological advances, among which we can quote mainly cyber physical systems (CPSs), internet of things (IoT), and internet of services (IoS). IoS enables service vendors to offer their services via the internet. The IoS consists of participants, an infrastructure for services, business models and the services themselves. Simple services are offered and combined into value-added services by various suppliers; they are communicated to users as well as consumers who can access them via various channels [3]. IoT represents an intelligent ICT infrastructure that allows real-time communication and cooperation between objects (machines and devices) as well as the connection between the physical and virtual world in a dynamic environment. CPSs are integrations of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa [4].

Based on the definition of CPS, "things" and "objects" can be understood as CPS. Therefore, the IoT can be defined as a network in which CPS cooperate with each other through unique addressing schemas [5]. The CPS thus allows an instantaneous and continuous interaction between physical elements and virtual elements and also with external actors. Objects become more and more autonomous and systems are increasingly reconfigurable because access to information becomes possible at any time and everywhere. This is multiplied by the intelligence capabilities given to objects and systems, or the ability to solicit through communication skills remote services able to provide this intelligence.

In this context, 'Industry 4.0' designs the factory, or even the entire production chain, as a big CPS largely self-regulated by machine-object interactions. Flexibility and agility, time, quality and cost earnings are expected [2]. This refers to the cyber-physical production systems (CPPS) concept. Indeed, relying on the latest and foreseeable developments of computer science, information and communication technologies, as well as manufacturing science and technology, CPPS may lead to the 4th industrial revolution ('Industry 4.0') [6].

An inevitable consequence of the ongoing transformation of the manufacturing world is the evolution of the logistics sector's organization. The objective is to allow a better pooling of logistics resources through the interconnection of several logistics actors for better performance in terms of efficiency and quality of services. If the vision of 'Industry 4.0' is to be realized, most enterprise processes must become more digitized.

A critical element will be the evolution of traditional supply chains toward a connected, smart, and highly efficient supply chain ecosystem [7–9]. With digitization, the supply network becomes an integrated ecosystem that is fully transparent to all the actors involved, from suppliers of raw materials and components to the transporters of those supplies and finished goods, and finally to the customers demanding fulfilment of their orders. This evolution led to the concept of cyber-physical logistics system (CPLS), which are based on the convergence of different scientific research topics, such as the operationalization of the holonic paradigm and/or the framework provided by the concept of physical internet.

In this paper, the authors propose to consider the CPLS concept for the physical internet deployment, based on the holonic paradigm. For this, the next section will provide an overview of concepts and paradigms related to this research question. Next, a CPLS model, based on isoarchic architecture of a global logistics system's control and on multicriteria, self-organized and dynamic decision-making, is proposed and justified. Then, the demonstrator allowing validating this proposal is presented. Finally, conclusions round up the paper and suggest future developments.

2 New Context of Global Logistics Systems

The CPLS is the result of CPPS applied to logistics. The CPPSs are themselves CPSs implemented in the context of manufacturing. According to [2], CPS brings together embedded systems that use sensors to retrieve data and act on physical processes by means of actuators. They are connected to each other via digital networks, use any data and services available worldwide, and benefit from multimodal human-machine interfaces. CPSs are therefore derived from the multi-technological integration, which has since long been referred to as the mechatronic, for 'mechanic + electronic'. This term, often deflected in research papers concerning only one of these two technologies, has not sufficiently emphasized the new capacities that real-time embedded electronics provide to operation systems in terms of capacities for perception, treatment and action. However, the distribution of computing functions and the digital processing introduced in operation systems with embedded communication capabilities open the classic mechanics and electronics framework. Indeed, this allows to envisage distributed systems whose components are sufficiently intelligent to give them autonomy capabilities and hence possibilities for self-organization. This is what had been foreseen and exposed in [10].

The explosion of interest in CPS is firstly correlated with the ability to embed a real-time processing capability that allows both strong interactions with the CPS user and full real-time consideration of the environment allowing intelligent automation of the CPS. This gives the possibility to create new products, able to embed functionalities that give them marketing attractiveness justifying the extra cost compared to more conventional products.

Indeed, the emergence of new ICT concepts has profoundly transformed industrial systems' research [11, 12], providing the opportunity for a range of diverse proposals that previously seemed impossible to emerge. Authors of [1] have thus pointed out four big challenges addressing the following questions:

- How to allow data and information analysis, mining, integration and sharing?
- How to offer contextual capabilities in complex business environments?

- How to deliver new and intuitive ways for interacting with EIS?
- How to support the development of professional competences triggered by new scientific and technological advances?

We are very interested by the second point, because it is the one that can generate value, and that will make the success of CPLSs.

We are totally convinced that the distribution of intelligence and decision-making capability brings many advantages in the case of dedicated CPSs to production systems or logistics systems.

Many research teams have extensively studied this paradigm in advance. Authors of [13] give a picture of the various works carried out in France regarding IMS (Intelligent Manufacturing Systems). Monostori et al. [14], Leitão [15] and Pujo [16] have proposed a panorama of distributed control solutions, based on multi-agent systems and/or on the holonic paradigm.

Many studies have tried to capture these concepts, going beyond the simulations carried out via multi-agent platforms and using proven technologies such as WSNs (wireless sensor networks). For example, authors of [17] demonstrate how CPS applications exploit the physical information collected by WSNs to bridge real and cyber spaces and identify important research challenges related to CPS designs. Authors of [18] propose to build CPS for manufacturing and in-house logistics, using WSN component; this constitutes an instance of CPPS.

CPPS consist of autonomous and cooperative elements and subsystems that are getting into connection with each other in situation-dependent nodes, on and across all levels of production, from processes through machines and production systems up to production and logistics networks [19]. Three main characteristics of CPPS were underlined in [20]:

- *Intelligence* (smartness), i.e. the elements are able to acquire information from their surroundings and act autonomously.
- *Connectedness*, i.e. the ability to set up and use connections with other elements of the system—including human beings—for cooperation and collaboration, and with the knowledge and services available on the Internet.
- *Responsiveness* towards internal and external changes.

CPPS partly breaks with the traditional automation pyramid [21]. CPPS are associated to the 'collaborative automation' paradigm, i.e. interoperable network-enabled collaboration between decentralized and distributed embedded devices and systems. Authors of [22] propose the very similar concept of Industrial CPS, and identify 27 key challenges for its deployment, by assessing their difficulty, priority and maturity perspectives. This analysis can be extrapolated to the CPLS.

The concept of CPLS was introduced in a Chinese national journal [23]. Based on the characteristics of existing logistics system and CPS, the architecture and key technology of CPLS are here presented. The major challenges of CPLS research, from the aspects of system modelling, large-scale information acquisition, optimization, control and standardization are discussed.

At the same time, the notion of complementarity between 'Industry 4.0' and logistics activities became obvious. Authors of [24] present real world industrial components (such as: high rack system, conveyor belts and sorting systems) of an in-house logistics system. These components are modelled and operated in a multi-agent framework, which provides an insight for the development of material handling equipment using CPS components which can be adjusted flexibly and modularly.

A CPLS is a special form of a logistics system which describes the boundaries, elements and connections of a CPS in logistics [25]. In this work, CPLS is interpreted as follows: "A cyber-physical logistics system is to be understood as a summarization of primary cyber-physical systems which carry out logistics tasks. The CPSs are embedded, linked systems, which act and interact with the environment autonomously. In particular, logistics tasks deal with the flow of information and goods in the value chain. CPLS aspires to economic, ecological and social aims."

Authors of [26] describe how CPS in combination with logistics models can improve production planning, control and monitoring; in [27] a CPLS is developed in 'Industry 4.0' environment, applied to tire manufacturing. The main research question of this work is: Why and how can improve CPLSs a manufacturing environment?

CPLS induces the emergence of new economic models such as the Physical Internet [28], which is based on the metaphor of the Digital Internet: the idea is to learn from the transmission of information in order to imagine a similar transport of physical objects and, by ensuring a high service quality, to assure economic efficiency and promote sustainable development. The Physical Internet (PI) will lead to the development of open systems connecting the physical objects to the global Internet. This concept implies significant organizational changes and evolutions. CPLSs subscribe to this perspective. In [29] the basis of a CPLS is defined, by proposing a generic approach for determining the product transport logistics chain along the entire product path, by exploiting the concept of Physical Internet. Note that these approaches based on self-organization of smart logistics entities rely on infotronics technologies for their implementation; therefore, they lead to the implementation of CPLS.

3 Holonic Paradigm for CPLS

In [30] the holonic paradigm for the study of social systems is defined. The concept of 'holon' corresponds to an identifiable part of a system with a unique identity, being composed of subordinate parts and being at the same time part of a larger whole. This allows revisiting the notion of Open Hierarchical Systems Theory: any system must be observed as a multi-level hierarchy of semi-autonomous sub-sets forming vertical trees and horizontal networks.

In the context of the global IMS initiative, 1990, many research teams have worked to apply the holonic paradigm to the industrial production world. The few invariants resulting from the many holonic manufacturing system (HMS) proposals will be further highlighted in order to draw the contours of holonic modelling of CPLS.

3.1 Typical Interacting Entities in a CPLS

Authors of [16] have inventoried a large number of HMSs. It results that the lowest common denominator between these HMSs, often extremely different, lies in a core of three types of holonic entities. They are related to: (i) each object or thing being processed by the production system, (ii) each equipment involved in this treatment, and (iii) each treatment giving the characteristics of the mission or the associated task.

These three types of holons have different designations depending on the HMS, as well as on the specific operation modes, but the most cited models, such as PROSA [31], often refer to the set {Product, Resource, Order}. This principle remains valid in the context of logistics activities, and we can transpose these types of holonic entities into a Holonic Logistics System [29].

The problem becomes hen the following: to reach a destination, there are several ways; each way consists of a succession of segments. All the segments constitute the transport network. There are potentially several resources available to transport along a given segment in a given time slot. The connection between the successive segments is done in a hub, with a possible change of transport resource [29]. It is therefore necessary to choose all the resources on the different segments of a path to ensure the delivery date of a product, or at least a minimum delay.

If this problem seems to correspond to a classical optimization problem, this type of theoretical solutions is not really adapted, mainly for different reasons. First, at a global logistics system's scale of PI type, the size of the problem is significant and the times to obtain a solution are quite long. Moreover, the model of the problem changes for each new transport. The dataset of times slots should therefore be fixed in order to obtain an efficient solution. Finally, transport missions have a long duration; they depend on the speed of the transport resources used. Therefore, there is always possibly that randomness and perturbations occur, so that the initial optimal solution proves to be much less efficient in time. All these aspects inhibit reactivity and possibly the proposed objectives.

This is why we prefer to study self-organization solutions, based on opportunistic collaborations aiming at a good compromise among the specific objectives of the holonic entities in interaction. To do this, our vision of holonic modelling and associated behaviours for each of these three types of holons will be detailed.

3.2 Product, Resource, Order for Physical Internet

The **Product Holon** (PH) models the object to be transported. As in PROSIS [32], there is a PH_i for each transported object, regardless of its nature. It may be quite possible that a PH represents a batch of products, if all the products of this batch have the same destination and are associated with the same package. This is especially important for small and low value products. Apart from its intrinsic characteristics, a PH will be characterized by its dimensions, which will allow finding the ad hoc container for its transport, at its final destination and at its required delivery date. This holon has an apparently passive role, but it contributes to the performance evaluation (for example in terms of individual quality of service) acting as a whistle blower in case of under-performance (e.g. delivery date may be exceeded).

The **Resource Holon** (RH) models the transport means. At each means (truck, boat, train…) corresponds one RH. Each RH_j has the objective of maximizing its capacity; this is clearly within the scope of the physical internet objective—to reduce empty traffic. This scope can only be achieved within the framework of global resource pooling, by clearly separating the concepts of operators and freight forwarders, and therefore by equitably distributing the benefits between the different actors in proportion to their costs and added value. This principle allows using the available resources, with the unique concern of their operational profitability.

The **Order Holon** (OH) is the one that ultimately manages the actually carried out transport. In fact, each OH_k is carried by the PI-container [33, 34] which is physical item used to encapsulate physical objects. It inherits the associated PH_i characteristics, namely the place and date of delivery. Its objective is to partner with other PI-containers to constitute a container corresponding to the standard sizes of handling in Hubs. Indeed, at the heart of the physical internet operation, the aggregation/disaggregation of PI-containers on transport segments allows obtaining economies of scale and achieves better profitability for each resource use. This is where all the interest of the physical internet lies.

When associating two or more PI-containers, during encapsulation or composing relations, the set of concerned $\{OH_k\}_m$ is then represented by an OH_m. This OH_m can be one of the $\{OH_k\}_m$ which takes on the role of representing the other OH. At the arrival on a Hub, the OH_m checks to see if the $\{OH_k\}$ continues to go in the same direction, i.e. if they are going to transit on the same segment S. If it is the case, the OH_m searches a transport resource for $\{OH_k\}$; otherwise, it requests the Hub to initiate a procedure of disaggregation of the OH_m in order to orientate each OH_k to its corresponding segment.

Once new OH_x have been reconstituted, each of them starts a procedure to find an ad hoc transport resource. Many research works have already been carried out on this type of problem, as for example [35–38]. However, it is the behaviour of the holons in the hubs that drives the fluidity of the flows and the smoothing of the loads, on which depends the achievement of overall logistics system performance and durability [39], especially sought in the physical internet context [28].

3.3 Behaviour Rules of Holons in a Hub

The emergence of a control solution in a holonic system results from the interactions that these holons have managed. In our proposal, the set of RH and OH serves the PHs, each of them providing a part of the transport specifications to be carried out (place and delivery date).

OHs, which encapsulate a PH or aggregate a set of OHs, will therefore collaborate with RHs to develop a common solution, while having their own goals. An RH_j will inevitably seek to fill its transport capacity before its departure date $t_{dep,\ l,\ j}$ on a given segment S_l, whereas an OH_k shall aim at advancing the PH_i, which it is in charge of on a path that brings it closer to its destination and assures delivery on time.

Let us consider, $W_{a,b}$ the way (path) from the departure Hub H_a to the destination Hub H_b. This way is a given sequence of segments S_l such that $W_{a,\,b} = \bigcup_{l=1}^{n} S_l$ with:

- S_l: the segment between the departure Hub H_{la} and the destination Hub $H_{l,b}$; let $S_l = [H_{la}, H_{l,b}]$.
- $H_{1,a} = H_a$: the departure condition.
- $H_{l,b} = H_{l+1,a}$: the condition of transport continuity in an intermediate Hub.
- $H_{nb} = H_b$: the destination arrival condition.

Each RH_j and each OH_k launches calls for proposals, to which the interested holons respond. The responses are analysed using a multicriteria decision aid that allows highlighting the most advantageous conditions.

Thus, an RH_j launches a transport service offer for the segment S_l with the departure date $t_{dep,\ l,\ j}$. For its part, each OH_k receives different offers of transport services that may be suitable. These offers differ in terms of departure dates, closeness to the final destination, remaining margin relative to the delivery date, price, sustainability of the used transport mode or even reliability and reputation of the company organizing the transport. Then, the OH_k carries out a multicriteria analysis, which allows it to classify these transport offers. The first classified offer makes the best compromise among its own requirements. Then, the OH_k attempts to reply to the concerned RH_j. This latter receives a set of responses. If this set of responses represents a PI-containers volume that exceeds its own capacity, it carries out a multicriteria analysis to rank the priority transport requests. Then, the RH_j selects the first *p* first responses that enable it to fulfil its own capacity, informs the corresponding OH_k that they have been selected, and informs the other OH_k requesters of the refusal of their request. The latter then re-launches the multicriteria search process of a RH, excluding RH_j from the set of possibilities.

Beyond these rules ensuring the nominal operation of the global logistics system, other rules can be envisaged allowing increasing flexibility and reactivity, like the solutions that were already tested in [40, 41]. Thus, it is possible for an OH to anticipate its transports in the forthcoming segments, by responding predictably to RH and by pre-committing itself to certain offers. Thus, if the OH receives a better offer from a new RH, it can release the previously chosen offer. Similarly, if a RH has an urgent need for capacity to prioritize an urgent transport, it can look at the accepted ones to select that one that would accept a new transportation means.

4 CPLS Modelling

The complex mechanisms described above constitute the foundations of intelligence associated to each holon. While these mechanisms are specific to each type of holon (PH, RH and OH) being also specifically parameterized for each holon, the operation principles are common. The principle consists in the fact that the global logistics system is a CPLS, itself recursively composed of CPLS, until it reaches the elementary CPLS which correspond to the simplest holons: product to be transported, basic transport resources (truck, train, container-carrier) and PI—Containers. The transition from the holonic paradigm to the operationalization of a CPLS is presented below.

4.1 Basic Implementation of a Holon in a CPLS

The literature agrees that a holon consists of a physical part and an information part (the body and the brain). In our work, we call them M_holon and I_holon [32]. The transition from a conceptual holon to an operational CPLS requires to associate with the physical part M_holon an embedded computer, having communication capability, which will manage the digital processing supported by the I_holon. Different platforms can potentially reach this requirement to realize demonstrators: WSN motes, Arduino, Raspberry PI, ESP8266, a.o.

The I_holon can thus perform the various required functions: a *finite state machine* (FSM—ensuring the core operation of the CPLS by specializing the operation according to the current stage in the previously described decision mechanisms), a *communication function* (interacting with the other I_holons, whether they are close or remote, using the relay of a server for example), a *remote services' call function* (for example, a web service able to carry out a multicriteria analysis or another web service able to find a way) or *local perception functions* (temperature measurement for a refrigerated I-container, measurement of position via an embedded GPS), etc. These different functions require real-time multithread processing.

Since the codes are relatively generic, the holons' recursion can be managed in the elementary CPLS. For example, the I_holon of the OH_m aggregating the set of holons $\{OH_k\}_m$ can be hosted by the embedded computer of one of the OH_k, operating with the same algorithms, but with the appropriate settings.

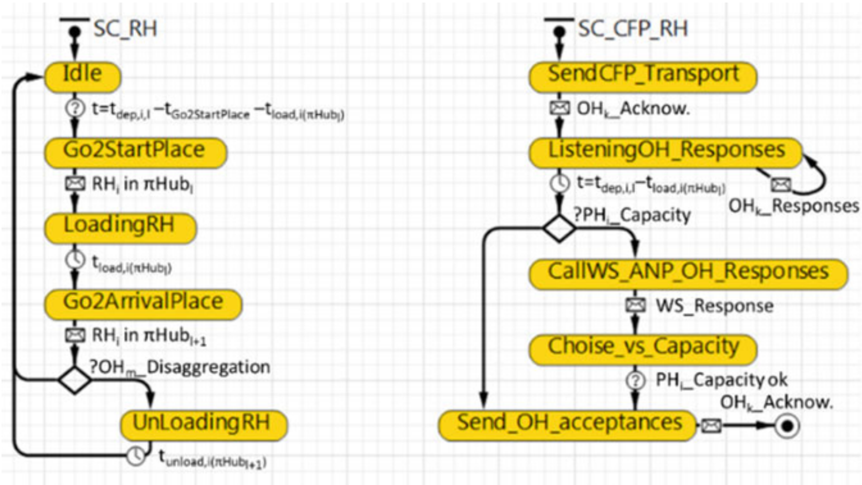


Fig. 1 Statechart diagrams of RH

4.2 *Statechart Diagrams*

All these different behaviours are formally modelled with Statechart diagrams [42]. Figure 1 shows two Statechart diagrams that operate in the FSM of a RH, in accordance with the behaviours described in Sect. 3.3: on the left, the transport control and on the right the calls for proposals manager.

4.3 *Multicriteria Decision Aid Used by OH*

The complex choices to be made within the CPLS are performed by a multicriteria decision aid via a web service (WS) offering an analysis based on ANP (analytic network process) [43]. Figure 1 shows the example of a call for such a web service for sorting and choosing the most interesting PI-containers to be transported for a RHi. Figure 2 shows an ANP structure aiming to choose the best possible path for an OH_k, i.e., the path that presents the best compromise between various criteria and indicators. The different alternatives A_q represent the different possible paths; the WS_ANP will classify them and the OH_k will try to follow the best possible way.

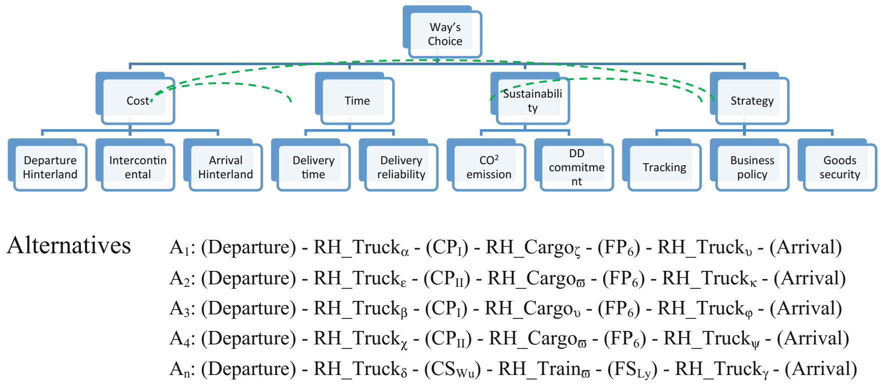


Fig. 2 ANP web service for OH

4.4 *Testbed: Intercontinental Exchanges China—France*

To illustrate the principles described above, we study the case of containers exchanges from China to France. We consider, on one hand, three Chinese economic and industrial zones (Boyal bay, Yangtze river delta, Pearl river delta) served by different Chinese Ports (CPi: Tianhin, Quigdao, Shanghai, Ningho, Hong Kong, Shenghen, Canton). On the other hand, two French Ports (FP) share the reception of maritime traffic (FP: Le Havre, Fos). It should be noted that the seaway (about 45 days) is in competition with the railway (Wuhan—Lyon: 15 days on average). The map in Fig. 3 shows the different way alternatives.

This work is in progress. We are looking for realistic data representing the different container flows. Two comparative studies will be conducted. In the first,

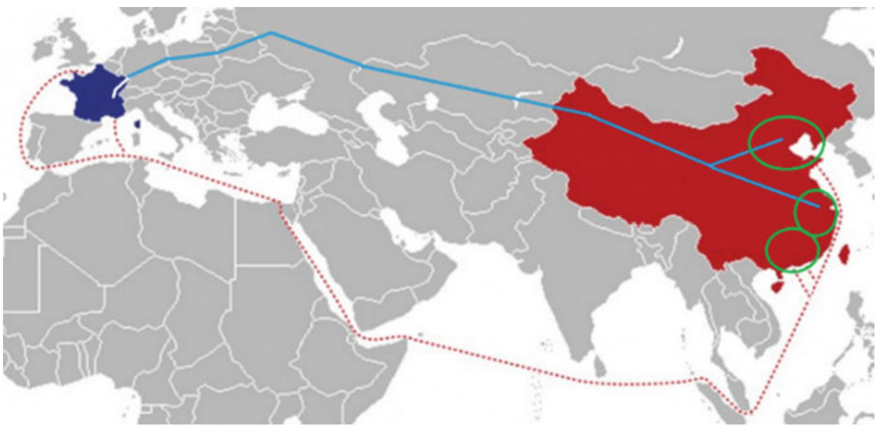


Fig. 3 Shipping and railway China—France

the flows will be positioned on fixed ways; then, seasonal variations will be introduced to identify peak loads and delays on performance indicators. In the second, with the same indicators and the same solicitations, we will observe that the flows will automatically balance and that the quality of the service will improve.

5 Conclusions

In this paper, a vision of the global logistics system of the future is presented, based on the Physical Internet, CPLS and the holonic paradigm. In this proposal, each holonic entity is a CPLS belonging to a more global CPLS; a CPLS is an implemented holon. Indeed, each physical entity is associated to an embedded computer which supports its associated intelligence. Each CPLS belongs to a network in which collective decisions emerge based on a multicriteria decision aid: ANP.

The next step is to set up different experimentations to highlight the advantages of this next logistics system generation. These experimentations will highlight great vulnerability to uncertainties in a static assignment phase and great behaviour robustness in a multicriteria and dynamic assignment phase. The decision structure will be enhanced and realistic data will be collected.

References

1. El Kadiri, S., Grabot, B., Thoben, K.D., Hribernik, K., Emmanouilidis, C., Von Cieminski, G., Kiritsis, D.: Current trends on ICT technologies for enterprise information systems. Comput. Ind. **79**, 14–33 (2016)
2. Kohler, D., Weisz, J.D.: Industrie 4.0: quelles stratégies numériques? La numérisation de l'industrie dans les entreprises du Mittelstand allemand, publication BPI-France Industrie 4.0, Kohler Consulting & Coaching (2015)
3. Buxmann, P., Hess, T., Ruggaber, R.: Internet of services. Bus. Inf. Syst. Eng. **5**, 341–342 (2009)
4. Lee, E.A.: Cyber-physical systems—are computing foundations adequate? NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Oct 16–17, Austin, TX (2006)
5. Hermann, M., Pentek, T., Otto, B.: Design Principles for Industrie 4.0 Scenarios, Business Engineering Institute St. Gallen, St. Gallen, Switzerland (2015)
6. Kagermann, H., Wahlster, W., Helbig, J.: Securing the future of German Manufacturing Industry: recommendations for implementing the strategic initiative industry 4.0. Final Report of the Industry 4.0 Working Group (2013)
7. Ounnar, F., Pujo, P.: Supplier evaluation process within a self-organized logistical network. Int. J. Logist. Manag. **16**(1), 159–172 (2005)
8. Schrauf, S., Berttram, P.: Industry 4.0: how digitization makes the supply chain more efficient, agile and customer-focused, Strategy and Pwc (2016)
9. Pan, S., Trentesaux, D., Sallez, Y.: Specifying self-organising logistics system: openness, intelligence, and decentralised control. In: Borangiu, T., Trentesaux, D., Thomas, A., Leitão, P., Oliveira, J-B. (eds.) Proceedings of Service Orientation in Holonic and Multi-agent Manufacturing (SOHOMA 2016), pp. 93–102. Springer International Publishing (2017)

10. Pujo, P., Ounnar, F.: Vers une approche holonique des Systèmes Mécatroniques Complexes: proposition d'un système de pilotage auto organisé et isoarchique. J Européen des Systèmes Automatisés **41**, 673–706 (2007)
11. Zhang, Y.F., Zhang, G., Wang, J.Q., Sun, S.D., Si, S.B., Yang, T.: Real-time information capturing and integration framework of the internet of manufacturing things. Int. J. Comput. Integr. Manuf. **28**(8), 811–822 (2015)
12. Trentesaux, D., Borangiu, T., Thomas, A.: Emerging ICT concepts for smart, safe and sustainable industrial systems. Comput. Ind. **81**, 1–10 (2016)
13. Cardin, O., Ounnar, F., Thomas, A., Trentesaux, D.: Future industrial systems: best practices of the intelligent manufacturing and services systems (IMS2) French Research Group. IEEE Trans. Ind. Inf. **13**(2), 704–713 (2017)
14. Monostori, L., Vancza, J., Kumara, S.R.T.: Agent-based systems for manufacturing. CIRP Ann.—Manuf. Technol. **55**(2), 697–720 (2006)
15. Leitão, P.: Agent-based distributed manufacturing control: a state-of-the-art survey. Eng. Appl. Artif. Intell. **22**(7), 979–991 (2009)
16. Pujo, P.: Pilotage isoarchique des systems de production. EME (2012)
17. Wu, F.J., Kao, Y.F., Tseng, Y.C.: From wireless sensor networks towards cyber physical systems. Pervasive Mob Comput **7**(4), 397–413 (2011)
18. Pujo, P., Ounnar, F., Power, D., Khader, S.: Wireless Holon network for job shop isoarchic control. Comput. Ind. **83**, 12–27 (2016)
19. Monostori, L.: Cyber-physical production systems: roots, expectations and R&D challenges. Procedia CIRP **17**, 9–13 (2014)
20. Monostori, L., Kadar, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., Ueda, K.: Cyber-physical systems in manufacturing. CIRP Ann.—Manuf. Technol. **65**, 621–641 (2016)
21. Vogel-Heuser, B., Kegel, G., Bender, K., Wucherer, K.: Global information architecture for industrial automation. ATP – Automatisierungstechnik **57**(1/2), 108–115 (2009)
22. Leitão, P., Colombo, A.W., Karnouskos, S.: Industrial automation based on cyber-physical systems technologies: prototype implementations and challenges. Comput. Ind. **81**, 11–25 (2016)
23. Lai, M.Y., Jin, C.F., Nie, K., Zhao, J.H.: Cyber physical logistics system: the implementation and challenges of next-generation logistics system. Syst. Eng. **4**, 008 (2011)
24. Lewandowski, M., Gath, M., Werthmann, D., Lawo, M.: Agent-based control for material handling systems in in-house logistics-towards cyber-physical systems in in-house-logistics utilizing real size. In: Proceedings of 2013 European Conference on Smart Objects, Systems and Technologies (SmartSysTech), p. 1–5 (2013)
25. Thoben, K-D., Veigt, M., Lappe, D., Franke, M., Kück, M., Kolberg, D., Fahl, I., Zimmerling, R., Schlick, J., Stephan, P., Guth, P.: Towards networking logistics resources to enable a demand-driven material supply for lean production systems—basic concept and potential of a cyber-physical logistics system. In: Proceedings of 7th International Scientific Symposium on Logistics, June 4–5, Bremen, pp. 45–71 (2014)
26. Seitz, K.-F., Nyhuis, P.: Cyber-physical production systems combined with logistic models—a learning factory concept for an improved production planning and control. Procedia CIRP **32**, 92–97 (2015)
27. Goedecke, K.L.: Development of a cyber-physical logistics system in an industrie 4.0 Environment, Master Thesis in Business Administration and Engineering, Univ. Hanover (2017)
28. Montreuil, B.: Towards a Physical Internet: Meeting the Global Logistics Sustainability Grand Challenge, Logistics Research, 2-/2011 (2011). https://doi.org/10.1007/s12159-011-0045-x
29. Ounnar, F., Pujo, P.: Holonic logistics system: a novel point of view for physical internet. In: Proceedings of International Conference on Information Systems, Logistics and Supply Chain (ILS 2016), June 1–4, Bordeaux (2016)
30. Koestler, A.: The Ghost in the Machine. Editions Hutchinson, London (1967)

31. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.: Reference architecture for holonic manufacturing systems: PROSA. Comput. Ind. **37**, 255–274 (1998)
32. Pujo, P., Broissin, N., Ounnar, F.: PROSIS: An isoarchic structure for HMS control. Eng. Appl. Artif. Intell. **22**(7), 1034–1045 (2009)
33. Landschützer, C., Ehrentraut, F., Jodin, D.: Containers for the Physical Internet: requirements and engineering design related to FMCG logistics. Logist. Res. **8**(1), 1 (2015)
34. Sallez, Y., Pan, S., Montreuil, B., Berger, T., Ballot, E.: On the activeness of intelligent Physical Internet containers. Comput. Ind. **81**, 96–104 (2016)
35. Pujo, P., Ounnar, F.: Proposition d'un pilotage décentralisé, auto-organisé et rapproché pour système automatisé flexible: Application à un Hub de transbordement robotisé fer-fer. J. Européen des Systèmes Automatisés **35**, 905–932 (2001)
36. Ballot, E., Gobet, O., Montreuil, B.: Physical internet enabled open hub network design for distributed networked operations. In: Proceedings of Service Orientation in Holonic and Multi-agent Manufacturing Control (SOHOMA 2012), pp. 279–292. Springer (2012)
37. Ballot, E., Montreuil, B., Thivierge, C.: Functional design of physical internet facilities: a road-rail hub. CIRRELT (2013)
38. Sallez, Y., Berger, T., Bonte, T., Trentesaux, D.: Proposition of a hybrid control architecture for the routing in a Physical Internet cross-docking hub. IFAC-PapersOnLine **48**(3), 1978–1983 (2015)
39. Frazzon, E.M. Sustainabily and effectiveness in global logistics systems. Ph.D., Univ Bremen (2009)
40. Ounnar, F., Pujo, P.: Pull control for job shop: holonic manufacturing system approach using multicriteria decision-making. J. Intell. Manuf. **23**, 141–153 (2012)
41. Ounnar, F., Pujo, P., Hachicha, M., Dubromelle, Y.: Study of an intelligent and multicriteria scheduling service, using academic benchmarks. Int. J. Comput. Integr. Manuf. **29**, 1166–1176 (2015)
42. Harel, D.: Statecharts: A visual formalism for complex system. Sci. Comput. Program. **8**, 231–274 (1987)
43. Saaty, T.L.: Decision Making with Dependence and Feedback: The Analytic Network Process. RWS Publications, Pittsburgh, PA (1996)

Simulation for PI-Hub Cross-Docking Robustness

Tarik Chargui, Abdelghani Bekrar, Mohamed Reghioui and Damien Trentesaux

Abstract Cross-docking is a logistics process which consists in receiving goods through unloading docks and then transferring them to the outgoing docks with almost no storage in between. A new concept named Physical Internet is applied to logistics based on the metaphor of the digital internet to improve the flexibility and synchronization of logistics systems. This paper presents a set of simulation scenarios of a real cross-dock facility of an industrial partner to compare the performances of the cross-dock with those of the PI-hub and to evaluate the contribution of the implementation of Physical Internet in cross-docking facilities. In this paper, two simulation models are proposed to compare performances of a classical cross dock and a PI-hub under the same flow of products and inter-arrival time of inbound and outbound trucks. Several scenarios are presented to test the robustness by changing the synchronization and the inter-arrival time distribution. To compare the performance of the cross-dock facility and the PI-hub, several key performance indicators (KPIs) are considered, such as waiting time, number of trucks waiting, resources usage and cycle time of the products.

Keywords Physical internet · Cross-docking · PI-Hub · Robustness · Simulation

T. Chargui (✉) · M. Reghioui
RSAID Laboratory, ENSATe, University of Abdelmalek Essaadi, Tetouan, Morocco
e-mail: tarik.chargui@gmail.com

M. Reghioui
e-mail: m.reghioui@gmail.com

A. Bekrar · D. Trentesaux
LAMIH Laboratory, UMR CNRS 8201, University of Valenciennes and Hainaut Cambrésis (UVHC), Le Mont Houy, 59313 Valenciennes, France
e-mail: abdelghani.bekrar@univ-valenciennes.fr

D. Trentesaux
e-mail: damien.trentesaux@univ-valenciennes.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_24

1 Introduction

1.1 Physical Internet (PI)

The Physical Internet (PI) is a new paradigm based on the digital internet, it was presented by Montreuil et al. [8] as a worldwide logistics network to make the current logistics systems more flexible and sustainable by changing the way physical objects are handled, moved, sorted and stored. The productivity and the speed of a supply chain have become an important factor of growth for industrial organizations and manufacturers. The Physical Internet aims to be an open system with an universal interconnectivity inspired from the digital internet by encapsulating physical objects in standardized PI-containers which are smart and modular containers that can be used to handle different freight volumes. In their paper, Montreuil et al. [7] provide the necessary elements for serving the Physical Internet infrastructure. These elements are classified on three categories: PI-containers, PI-movers and PI-nodes. The PI-containers are designed specifically for the Physical Internet to manipulate parts, raw materials or final products in different combinations of dimensions and size. These PI-containers are handled, routed and stored using the Physical Internet system through PI-movers, such as PI-trucks, PI-wagons, PI-conveyors and PI-lift-trucks. The PI-nodes are the locations which interconnect the logistics network of the Physical Internet. These PI-nodes can perform operations on PI-containers such as assembling, disassembling, picking, routing and monitoring.

Ballot et al. [2] presented a functional design that is feasible for a road-rail hub facility which is a PI-node that transfers the PI-containers from inbound train to outbound trucks and from a train to another train. The paper identifies also the ways to measure the key performance indicators (KPIs) of the proposed design. Meller et al. [6] proposed a road-based transit PI-transit that provides the mechanism to transfer PI-containers from inbound trucks to the outbound trucks, and presented also the design of the PI-transit facility and the KPIs to measure the performance of the facility from customers and PI-transit operators perspective.

Montreuil et al. [9] presented the functional design of a road-based cross-docking hub (Fig. 1) for the transshipment of the PI-containers from inbound to outbound trucks. The paper provides also the KPIs that can be used to measure the performance of the cross-docking hub. Walha et al. [12] provided a heuristic to solve a rail-road PI-hub allocation problem by allocating the PI-containers to the outgoing trucks and then improve the solution with a simulated annealing method. The paper takes in consideration the dynamic aspect by providing a multi agent approach to find a reactive solution. Sallez et al. [10] presented a hybrid control architecture for the routing of PI-containers in a PI-cross-docking hub to face external perturbations of conveying units failure.

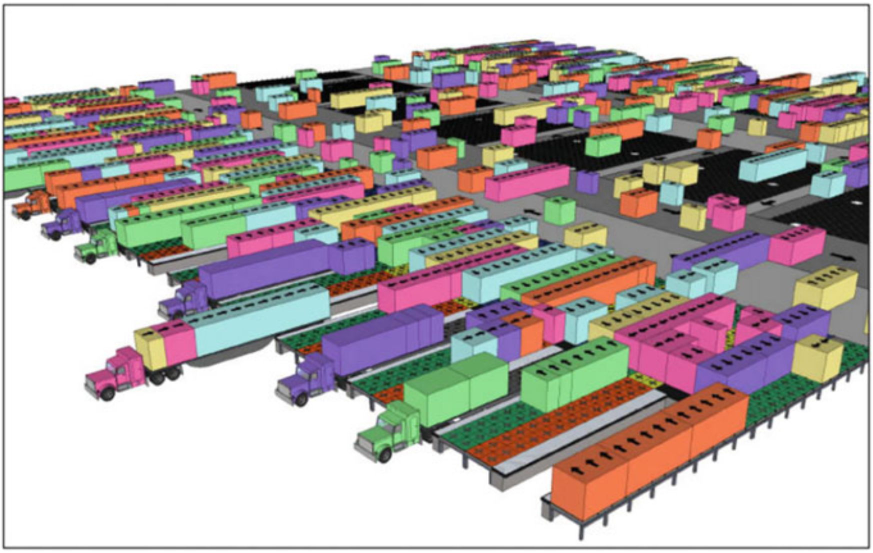


Fig. 1 Cross-docking hub facility layout [9]

1.2 Cross-Docking

The cross-docking is a logistics strategy which consists of unloading freight through inbound docks and loading them into outgoing trucks directly, or after passing through a temporary storage in between. Inside the cross-docking facility, the inventory can be kept for less than 24 h. van Belle et al. [11] presented a state of the art which addresses a variety of cross-docking problems that are classified from strategic, tactical to operational. Another state of the art was presented by Ladier and Alpan [5] with a similar framework for comparing the literature review by classifying the problems from strategic, tactical to operational. Authors give also performance measures (KPIs) and a classification of cross-docking problems into truck-to-door assignment, sequencing and scheduling.

This paper is an extension of the work presented in Chargui et al. [3]. It considers a real case of 5 inbound doors to 5 outbound doors cross-docking facility of an industrial partner. The objective is to compare the performance of a classic cross-dock and a PI-hub using KPIs such as cycle time, waiting time of inbound and outbound trucks and the resources utilization.

The remaining of this paper is organized as follows. In Sect. 2, the parameters of the studied cross-dock are presented with its resources and layout characteristics. Section 3 presents the simulation scenarios for the cross-dock and the PI-hub. Section 4 presents the analysis of the results obtained to compare the performance of the cross-dock and the PI-hub.

2 Cross-Dock Overview and Simulation Parameters

2.1 Cross-Dock Overview

In this simulation, a real case of an industrial partner cross-dock is considered. The cross-docking facility contains 5 inbound doors and 5 outbound doors managing 5 different types of household appliances products (type 1, 2, 3, 4 and 5) coming from five different suppliers.

The model considers an intermediate storage warehouse (Fig. 2) separated into 5 areas (ane area for each one of the five products). For the resources, 5 forklifts are considered (forklifts usage reaches 70% for a day with an important number of arrivals: 50 inbound trucks and 50 outbound trucks). During the simulation the forklifts are dynamically assigned to inbound and outbound docks and storage areas. For example, if there is a forklift that is not used it will be assigned automatically to the closest dock or storage area for support.

The quantity of products carried by an inbound truck is equal to 50 which is the truck capacity for this simulation model. The products are unloaded using the available forklifts and then transferred either to the outbound docks if there is a truck requesting that product or to the storage areas waiting for outbound trucks to arrive.

For the outbound docks, trucks arrive on each dock with a random demand of different products. In order to respect the FIFO rule, forklifts have to pick up products from the warehouse first. However, if the quantity of products in the storage area is not sufficient and there is a truck unloading that type of product in the reception

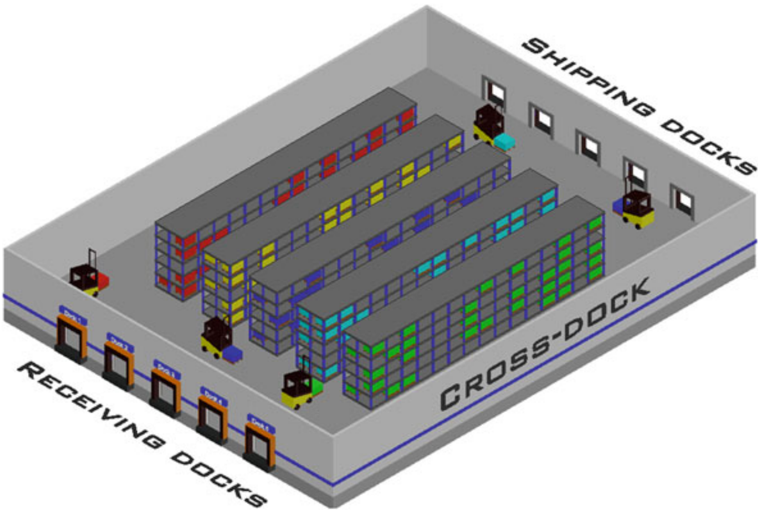


Fig. 2 An overview of the cross-docking facility layout

Table 1 Simulation parameters for the cross-docking facility

Parameters		Values
Number of docks	Inbound docks	5
	Outbound docks	5
Number of product types		5
Trucks capacity		50
Number of products requested by outbound trucks		Random
Average loading and unloading time		20 s
The average speed used for the forklifts		7 km/h
Cross-dock facility surface		6000 m^2
Number of trucks served in 24 h (Normal day)	Inbound trucks	25
	Outbound trucks	25
Number of trucks served in 24 h (Busy day)	Inbound trucks	50
	Outbound trucks	50
Forklifts operators pause duration		20 min
Time between the start of two successive pauses		4 h
Trucks inter-arrival time with sync. (in minutes)	Normal day	TRIA (230, 290, 350)
	Busy day	TRIA (115, 145, 175)
Trucks inter-arrival time without sync. (in minutes)	Normal day	EXPO (120)
	Busy day	EXPO (60)

docks then the requested quantity will be picked up directly from that inbound truck without passing through the storage area.

This simulation model takes into consideration the forklifts drivers pauses of 20 min every 4 h. The remaining simulation parameters are specified in Table 1.

2.2 PI-hub Cross-Docking

For the PI-hub cross-docking simulation model, the inter-arrival times for all the scenarios, the number of docks and the cross-dock facility surface are kept the same as the cross-dock model. However, instead of using forklifts, automated loading and unloading PI-docks connected to the storage area using a flexible network of high

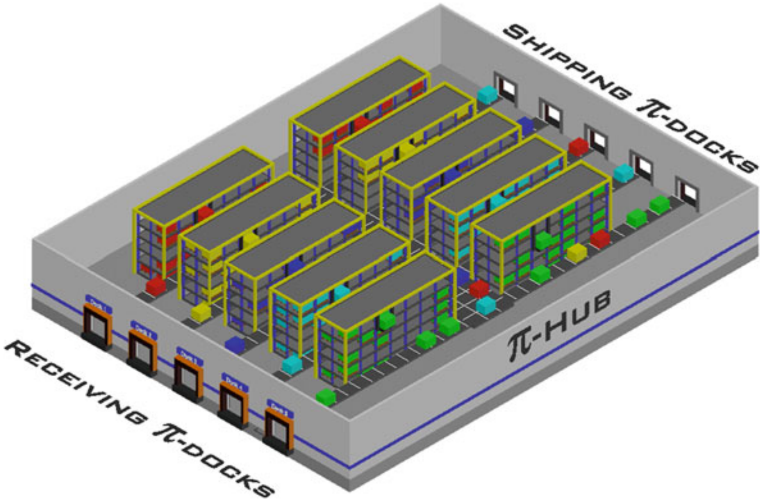


Fig. 3 An overview of the PI-hub facility layout

speed PI-conveyors are considered (average speed of 10 km/h). Moreover, an automated storage and retrieval system (AS/RS) in the storage area is connected to the PI-sorter in the sorting zone. An overview of the PI-hub cross-docking facility is presented in Fig. 3.

3 Simulation Scenarios

The two simulation models are developed using ARENA Simulation software [4], [1] (version 13.50). The simulation models were executed for 24 h, 10 replications for each scenario. Two inter-arrival distributions of inbound and outbound trucks are considered: Triangular (Min, Mode, Max) for the synchronized scenarios, and EXPO (Mean) for non-synchronized scenarios. The values generated by the triangular distribution are more controlled using the Min, Max and Mode as parameters, which returns values that are close to the Mode parameter. For the Exponential distribution, the values generated are less controlled because it takes only the Mean as parameter.

For each inter-arrival distribution, two scenarios are considered for each distribution: the first one considers a normal day of the cross-dock facility with 25 inbound trucks and 25 outbound trucks; the second scenario considers a busy day with 50 inbound trucks and 50 outbound trucks. All the distributions parameters used in these scenarios are summarized in Table 2.

In this paper, four KPIs are considered: first, the average and the maximum waiting time of inbound and outbound trucks spent at the docks for unloading/ shipping

Table 2 Inter-arrival distributions parameters used in the different scenarios

		Inter-arrival distribution (in minutes)		
		TRIA		EXPO
	Min	Mode	Max	Mean
Busy day	115	145	175	60
Normal day	230	290	350	120

products. For the inbound trucks, the waiting time is usually related to the availability of the forklifts, and for outbound trucks the waiting time depends on the forklifts and products availability which is related to inbound trucks delays. The second KPI is the number of inbound and outbound trucks waiting, which is an important indicator that helps designing the cross-dock facility layout, especially the docks doors queuing places (trucks waiting areas). The third KPI is the average and total time (cycle time) spent by a product in the cross-dock facility. The last KPI is the resources' utilization which includes forklifts for the cross-dock and conveyors utilization for the PI-hub.

4 Simulation Results and Analysis

4.1 Waiting Time

The waiting time considered in this paper includes the time spent by the truck waiting at the docks to unload or to load the products. Table 3 shows the deviation between the average and maximum waiting time in the cross-dock and the PI-hub. For each scenario the deviation between the PI-hub and the cross-dock is presented in the third row. The last column presents the total average of the deviation values. The deviation is improved especially for non-synchronized scenario in busy days, which shows the robustness of PI-conveyors network in case of perturbation compared to the forklifts that are shared with inbound and outbound docks and the storage area.

The following graph (Fig. 4) summarizes the average deviations of waiting time between the PI-hub and the cross-dock.

4.2 Number of Trucks Waiting at Docks

Table 4 presents, the average and maximum number of trucks waiting at the inbound or outbound docks for all the scenarios presented previously. As shown in Table 4, the number of inbound and outbound trucks waiting at docks was considerably

Table 3 Trucks waiting time at inbound and outbound docks

Trucks synchronization			Waiting time at docks				
			Inbound		Outbound		Total average
			Average	Max	Average	Max	
Sync.	Busy day	Cross-dock	42.7	120.0	83.9	339.8	
		PI-hub	16.5	98.3	16.5	323.8	
		Dev. (%)	−61.4	−18.1	−80.3	−4.7	−70.9
	Normal day	Cross-dock	36.4	178.5	65.1	373.8	
		PI-hub	16.5	161.2	16.5	345.2	
		Dev. (%)	−54.7	−9.7	−74.6	−7.7	−64.7
Non-sync.	Busy day	Cross-dock	154.4	414.0	358.5	991.2	
		PI-hub	19.3	174.2	30.2	489.8	
		Dev. (%)	−87.5	−57.9	−91.6	−50.6	−89.6
	Normal day	Cross-dock	51.1	331.0	126.4	862.4	
		PI-hub	17.7	297.3	21.2	796.5	
		Dev. (%)	−65.3	−10.2	−83.3	−7.6	−74.3
						Average	−74.8%

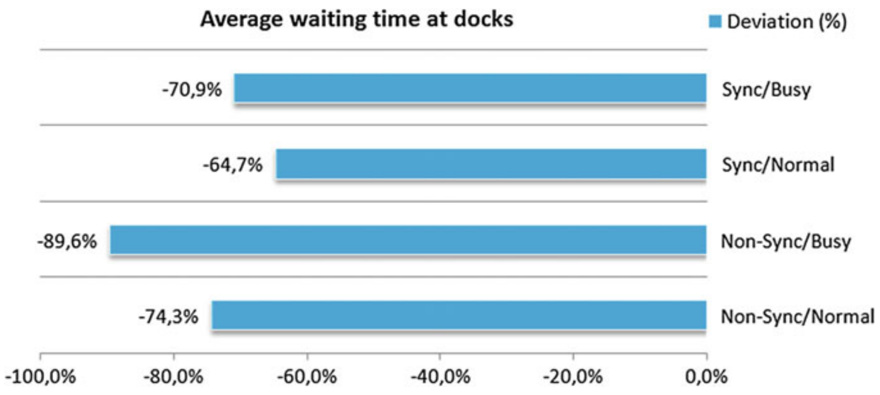


Fig. 4 Average deviation of trucks waiting time at docks

Table 4 Number of trucks waiting at inbound and outbound docks

Trucks synchronization			Waiting time at docks				
			Inbound		Outbound		Total average
			Average	Max	Average	Max	
Sync.	Busy day	Cross-dock	0.3	0.8	1	2.1	
		PI-hub	0.1	0.7	1	2	
		Dev. (%)	−60.50	−16.40	−2.00	−6.50	−31.25
	Normal day	Cross-dock	0.1	0.7	1	1.6	
		PI-hub	0.1	0.6	1	1.6	
		Dev. (%)	−53.90	−8.50	0.00	−3.70	−26.95
Non-sync.	Busy day	Cross-dock	1.3	3.6	4.4	7.6	
		PI-hub	0.2	1.9	2.2	4.9	
		Dev. (%)	−84.60	−47.70	−50.50	−34.90	−67.55
	Normal day	Cross-dock	0.2	1.5	1.9	3.4	
		PI-hub	0.1	1.4	1.5	3.1	
		Dev. (%)	−65.30	−10.00	−20.20	−7.10	−42.75
						Average	−42.13%

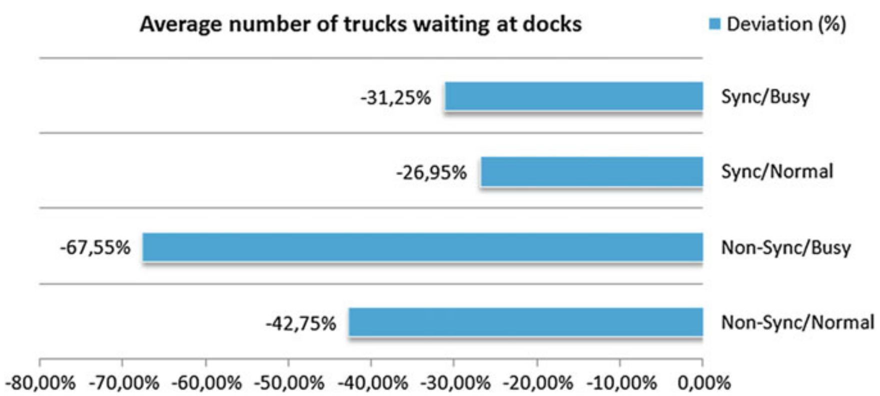


Fig. 5 Average deviation of number of trucks waiting at docks

minimized in the PI-hub configuration with a total average deviation of −42.13% particularly in busy days and when trucks inter-arrival times are not synchronized.

The following graph (Fig. 5) summarizes the average deviations between the PI-hub and the cross-dock.

4.3 Total Time

The total time of a product (or cycle time) in the cross-dock starts once the truck arrives at the dock and ends when the product leaves the cross-dock on an outgoing truck. In Table 5, as for the previous KPIs, the total time was significantly improved particularly for non-synchronized inter-arrival time in a busy day at the cross-dock facility.

A summary of average deviation total time is presented in Fig. 6.

Table 5 Products total time

Trucks synchronization			Total time	
			Average	Max
Sync.	Busy day	Cross-dock	146.8	473.8
		PI-hub	125.2	436.9
		Dev. (%)	−14.8	−7.8
	Normal day	Cross-dock	199.4	527.4
		PI-hub	182.1	500.2
		Dev. (%)	−8.7	−5.1
Non-sync.	Busy day	Cross-dock	338.3	752.0
		PI-hub	198.1	524.1
		Dev. (%)	−41.4	−30.3
	Normal day	Cross-dock	307.6	720.4
		PI-hub	284.6	699.3
		Dev. (%)	−7.5	−2.9
		Total average	−18.1%	

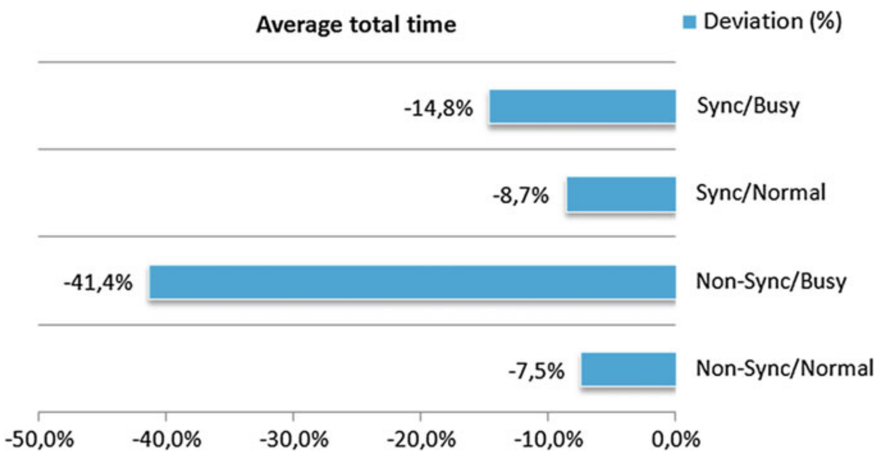


Fig. 6 Average deviation of total time

4.4 Resources Utilization

As presented in Table 6, the forklifts utilization increases in case of non-synchronized inter-arrival time particularly in busy days when the number of trucks increased. Conveyors utilization stays almost the same in all scenarios. The last column shows the average deviation between the forklifts and the AS/RS robots' utilization.

The average deviation of resources utilization is summarized in Fig. 7 for the different scenarios.

We can see clearly from the previous results that for the same workload, the utilization rate of PI-hub resources remains far under the one of classical cross docks. This is mainly due to the benefits of automation and the important handling capacity allowed by conveyors and AS/RS systems.

Table 6 Cross-dock resources utilization

Trucks synchronization			Resources utilization			Dev. (%)
			Forklifts (%)	Conveyors (%)	AS/RS robots (%)	
Sync.	Busy day	Cross-dock	71.90	–	–	−18.89
		PI-hub	–	0.10	58.32	
	Normal day	Cross-dock	40.10	–	–	−19.38
		PI-hub	–	0.00	32.33	
Non-sync.	Busy day	Cross-dock	89.80	–	–	−53.28
		PI-hub	–	0.10	41.96	
	Normal day	Cross-dock	50.00	–	–	−22.26
		PI-hub	–	0.10	38.87	
				Average deviation		−28.45

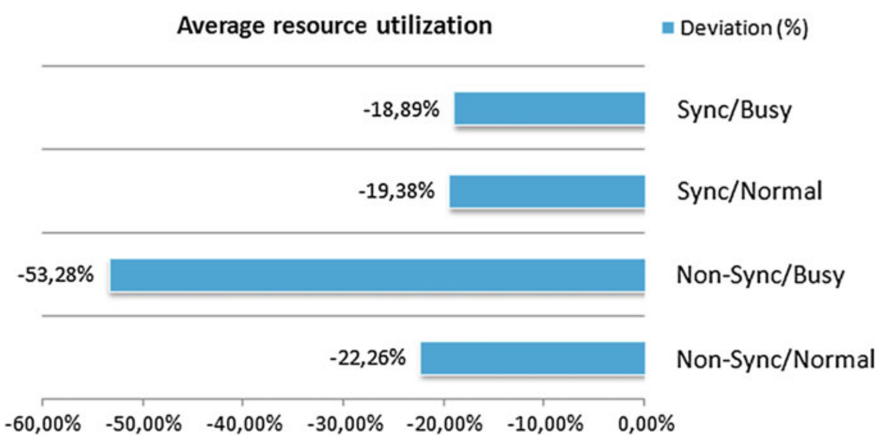


Fig. 7 Average deviation of resource utilization (AS/RS robots vs Forklifts)

5 Conclusion

To sum up, the simulation of the cross-dock and the PI-hub with a variety of scenarios shows that the implementation of Physical Internet has a positive impact on the performance of cross-docking facilities by reducing especially the waiting time of inbound and outbound trucks, the total time spent by products in the cross-dock and the number of inbound and outbound trucks waiting for a service. The simulation results show the robustness of the Physical internet configuration particularly in case of a bad synchronization between inbound and outbound trucks or an increase of the products flow in the PI-hub. In future work, more complicated configurations will be taken into consideration and more approaches will be developed for the new PI-hub cross-docking system to optimize the assignment of trucks to docks and jobs to resources.

References

1. Arena simulation website (2016). https://www.arenasimulation.com/
2. Ballot, E., Montreuil, B., Thivierge, C.: Functional design of physical internet facilities: a road-rail hub. In: Proceedings of the International Material Handling Research Colloquium (IMHRC), pp. 1–34 (2012)
3. Chargui, T., Reghioui, M., Bekrar, A., Trentesaux, D.: A simulation model to migrate from cross-docks to pi-hubs. In: The 3rd International Physical Internet Conference (IPIC 2016), pp. 1–11. Atlanta, GA, USA (2016)
4. Kelton, W., Sadowski, R., Sadowski, D.: Simulation with ARENA. McGraw Hill (2000)
5. Ladier, A.L., Alpan, G.: Cross-docking operations: current research versus industry practice. Omega **62**, 145–162 (2016)
6. Meller, R.D., Montreuil, B., Thivierge, C., Montreuil, Z.: Functional design of physical internet facilities: A road-based transit center. In: Proceedings of the International Material Handling Research Colloquium (IMHRC), pp. 1–32 (2012)
7. Montreuil, B., Meller, R.D., Ballot, E.: Towards a physical internet: the impact on logistics facilities and material handling systems design and innovation. In: Proceedings of the International Material Handling Research Colloquium (IMHRC), pp. 1–23 (2010)
8. Montreuil, B., Meller, R.D., Ballot, E.: Physical internet foundations. In: IFAC Proceedings Volumes, pp. 26–30 (2012)
9. Montreuil, B., Meller, R.D., Thivierge, C., Montreuil, Z.: Functional design of physical internet facilities: a road-based crossdocking hub. In: Proceedings of the International Material Handling Research Colloquium (IMHRC), pp. 1–34 (2012)
10. Sallez, Y., Berger, T., Bonte, T., Trentesaux, D.: Proposition of a hybrid control architecture for the routing in a physical internet cross-docking hub. In: IFAC, pp. 1978–1983 (2015)
11. van Belle, J., Valckenaers, P., Cattrysse, D.: Cross-docking: state of the art. Omega **40**, 827–846 (2012)
12. Walha, F., Chaabane, S., Bekrar, A., Loukil, T.: A rail-road pi-hub allocation problem: active and reactive approaches. Comput. Ind. **81**, 138–151 (2016)

Part VI
Formal Methods and Advanced Scheduling for Future Industrial Systems

A New HMI Scheduling Model Implemented on a Real Manufacturing Scheduling System

Zakaria Yahouni and Nasser Mebarki

Abstract Manufacturing scheduling systems operate in a highly dynamic environment where several perturbations occur during the execution of a schedule. These perturbations prevent the execution of the planned schedule exactly as it was supposed to. To cope with this drawback, some techniques propose several scheduling solutions instead of a unique one, allowing during the execution phase to select the appropriate schedule that hedges against the perturbations. This selection is effectively done by a human operator who plays a key role in decision-making because of his knowledge and expertise. But, because of the limited complexity that he can handle, he needs to cooperate with the machine to take efficient decisions. In this context, the Human-Machine-Interface (HMI) literature research in planning and scheduling is rather theoretical. This paper addresses this relation from a practical-oriented perspective by proposing a new HMI model adapted to the *groups of permutable jobs* method, which is one of the most used literature methods to cope with shop perturbations. A practical experiment using the proposed HMI is then conducted. The results stress the usability and the limits of the proposed model.

Keywords Manufacturing scheduling · HMI · Decision support

1 Introduction

In an industrial production context, planning designates generally the *off-line production phase*, where tools such as Gantt chart are used to model and plan the production activities, whereas scheduling generally refers to the *online production*

Z. Yahouni (✉) · N. Mebarki
LS2N, Laboratoire des Sciences du Numérique de Nantes, UMR CNRS, 6004 Nantes, France
e-mail: zakaria.yahouni@ls2n.fr

N. Mebarki
e-mail: nasser.mebarki@ls2n.fr

Z. Yahouni
MELT, Manufacturing Engineering Laboratory of Tlemcen, Tlemcen, Algeria

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_25

phase where production activities are executed in real-time on the available resources based on the off-line planned schedule [1].

During the online phase, a simple perturbation such as *machine breakdown* prevents the execution of the off-line schedule exactly as it was planned and therefore deteriorates its expected performances. To cope with this drawback, several techniques seek to incorporate models of disturbances into the off-line schedule in order to absorb shop floor uncertainties [2, 3]. These techniques usually construct a flexible off-line schedule that encapsulates a family of schedules instead of a unique one, such that during the online phase the schedule that copes appropriately with the current perturbation is selected in real-time. This selection is usually done either using a reactive-scheduling-algorithm policy or by a decision-maker: *human operator* who is more capable for adapting to the shop floor volatility [4].

In this context, the human appears to be crucial for taking the large set of scheduling constraints into account and adapting to changes [5, 6], but a human operator is limited by the complexity of large calculations that are usually handled easily by machine algorithms. Therefore, a human-machine cooperation is critical for the scheduling decision process.

To our knowledge, most of the literature works on human-machine cooperation in manufacturing scheduling are rather theoretical. In this paper, we investigate the cooperation from a practical-oriented perspective by conducting a case study using a real manufacturing system. The proposed contribution consists of a new HMI that identifies the relation and control between the human and the machine.

The proposed HMI is driven by a *decision support system* (DSS) that incorporates three decision-aid criteria. This DSS should favour the activity of the operator and therefore the scheduling performance.

In Sect. 2, a literature review on the human-machine cooperation in scheduling is given. Section 3 is devoted to the proposed HMI and its application to the groups of permutable jobs. Then, a case study is presented in Sect. 4. This section contains also the experimental protocol and results concerning the usability of the proposed HMI. Final conclusions are summarized in the last section.

2 Literature Review

The human intervention in planning and scheduling has been a conflicted issue between operation research researchers and artificial intelligence experts since the fifties. Observations of the human decision behaviour in scheduling were first formulated by [5]. Artificial intelligent researchers claim that such behaviour is algorithmic and can be replaced by an expert system [7, 8]. However, the work of decision-support experts revealed that the human decision behaviour is non-algorithmic and differs somewhat from rational theories; because of certain factors that occur in real-time and can not be taken in advance by an algorithm [6, 9, 10].

In this context, various studies have been made on the basis of observation analyses of the human behaviour [6, 11]. These studies had the scope of investigating the factors affecting the human behaviour in a real scheduling environment. Three main factors were identified:

- Factors related to the human natural brain, including the social environment, biological formation, cognitive, psychological mindset and intuitive judgment.
- Organizational factors such as the organization and management of the different activities interfering with its primary scheduling activity.
- Technological factors such as interfaces and software used during the decision-making scheduling process.

The work presented in this paper focuses on the technological factors affecting the human scheduling decisions. In this context, some preliminary questions about the organizational context need to be addressed. These questions focus primarily on the amount of scheduling control and role given to each entity: the human and the machine. The work of [12] characterizes various control levels:

- Manual control: all the decisions are made by the human and the machine has no control at all.
- Advisory control: all the decisions are made by the human, but the machine checks the feasibility of each decision.
- Interactive control: the decisions are shared between the human and the machine.
- Supervisory control: in this mode, the machine is playing the role of a supervisory control for taking decisions, but the human validates them.
- Automatic or algorithmic control: contrarily to the manual control, all the decisions are made by the machine algorithms which replace the human.

These control modes allow determining the decision task of the two entities: the human and the machine. In a manufacturing context, the aim is not to exclude any entity such as in manual and automatic control, but to exploit the cooperation between the human thinking and the machine high computation complexity in order to achieve maximum scheduling efficiency.

The identification of the control mode leads to the definition of the HMI. However, designing a HMI model can follow general theoretical principles based on theories of cognitive processing, ergonomics and technological interfaces [13–16]. The work of [16] addresses the human decision process for the time-table problem. The authors stress the lack of ergonomic and ecological HMI software in order to take advantage of the human-machine cooperation.

Furthermore, the research perspectives on HMIs have a particular interest in scheduling. Most of the proposed HMI literature are based on observational analysis in real or simulated situations using realistic scenarios. The resulted HMIs emphasize the usefulness of a DSS in order to assist the human decisions.

A DSS has been proposed in [17] in order to optimize and manage the transportation scheduling problem. A set of scheduler-oriented algorithms taking into account operational and cognitive research aspects has been integrated into the proposed HMI. These algorithms are developed based on a 3-phase approach and use the three cooperation control modes in order to favor the human-machine cooperation for the transportation scheduling problem. The effectiveness of the proposed DSS is illustrated on small size benchmark instances.

Furthermore, another recent study presented in [18] has led to the definition of a new industrial HMI model called i-DESMÉ. This model permits the creation of decision support software to assist the decision scheduling process in micro/small companies. The advantage of the proposed model is that it considers multidisciplinary characteristics of the HMI related to the human behaviour factors, such as ergonomics and cognitive aspect. The proposed model has identified several oriented decision algorithms that provide support to the human scheduler during the online phase of the scheduling process.

3 A Proposed HMI Using Groups of Permutable Jobs

In this section, we propose a new HMI for scheduling under uncertainties using the *groups of permutable jobs* method which is one of the most studied proactive-reactive approaches to cope with uncertainties. The following section gives a brief theoretical background about this method.

3.1 Groups of Permutable Jobs

The method was first introduced in [19] and has been implemented in an industrial software called *ORDO* [20, 21], which has been used by more than 70 make-to-order manufacturing companies in France.

This method provides a flexible solution to the scheduling problem, that encapsulates a set of schedules during the planning phase. The groups of permutable jobs are used in two stages:

- The first stage consists of proposing an off-line flexible schedule that represents a family of schedules instead of a unique one; the technique used merges activities (jobs) executed successively on the same resources (machines), such that any order of jobs within each group leads to a feasible schedule that can be executed in the shop.
- The second stage is a decision stage. It consists of executing in real-time one of the schedules characterized by the groups of permutable jobs, and therefore choosing the next job to proceed within each group.

These two stages are illustrated on a flow shop scheduling example. In a flow shop problem, we have a set of jobs $J_1, J_2, ..., J_n$ executed on the same set of machines $M_1, M_2, ..., M_m$. A scheduling solution to this problem consists of ordering the jobs in each machine. The performance of this solution is usually measured through a regular objective function. In this paper, the performance is measured using the makespan

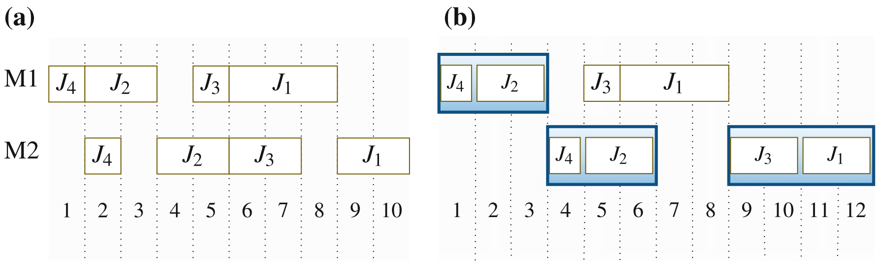


Fig. 1 Example of groups of permutable jobs

performance which refers to the maximum completion time of all jobs (C_j is the completion time of J_j and C_{max} is the maximum completion time of all jobs).

A group of permutable jobs in a given machine represents a group of at least two jobs that can be executed in an arbitrary order during the second stage. Figure 1a and Fig. 1b illustrates respectively a solution of a four-jobs/two-machines flow shop and a flexible schedule represented by groups of permutable jobs. The flexible schedule constructed in stage 1 contains three groups of permutable jobs. These groups characterizes eight different schedules.[1] The decision phase consists of selecting the order of jobs in each group of permutable jobs. At the end, one of the eight possible schedules is established in the shop.

3.2 Decision Stage of the Groups of Permutable Jobs

To allow the human-machine cooperation, we present in Fig. 2 the overall model for the decision stage of the groups of permutable jobs. The operational concept of this model requires that the human interacts with the machine using a DSS (called Decision-aid System in the figure) and that during the online phase of the groups of permutable jobs.

The DSS represents the core of the proposed HMI and is supposed to assist the operator with all the information needed during the decision phase. This DSS is composed of three modules:

- The *decision-aid interface*, which represents the interlocutory module between the human requests and the machine assistance. It allows providing the operator with the information needed and also triggering the jobs' execution according to the order selected by the operator.
- Data storage: this module stores all the variables of the planned off-line schedule such as the jobs starting time, completion time, etc.

[1]The number of schedules is the factorial multiplication of the number of jobs in the groups: (2! × 2! × 2!).

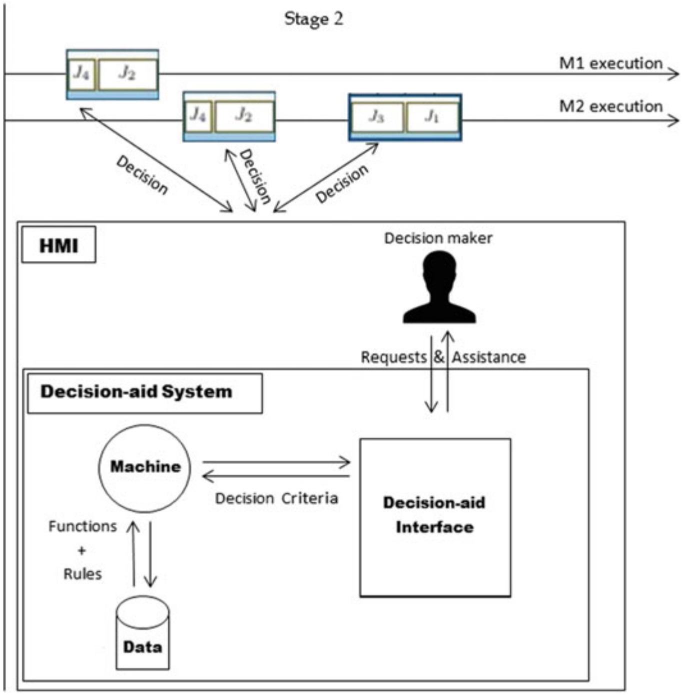


Fig. 2 Proposed HMI model

- The machine uses the information stored in the data module in order to compute and assist the requests of the decision-maker.

In this proposed HMI, we assume that the human has the control over all decisions, and the machine evaluates, validates and assists his decisions; because it is more useful to calculate the consequences of the human decisions by the machine rather than the human itself. In addition, as stated by [4], the human scheduler remains efficient for taking decisions, although not optimal, contrarily to the machine which is optimal in computation, but with relatively poor performance in realistic situations.

However, the requests/answers transferred between the decision-maker and the decision-aid system are carried out using a simple decision-aid interface. In this interface, the decision-maker finds all the information concerning each job of a group so that he can make correct/optimal decisions. According to the information provided, the decision-maker may or may not take advantage of the decision-aid interface, which can significantly increase or reduce its activity and therefore affects the scheduling performance as stated by [22] who studied the activity of manual rescheduling. In order to avoid this lack of interaction, we propose to use three decision-aid criteria in order to anticipate the consequences of a decision before taking it:

- *Best-case schedule*: it provides the best possible performance if the selected job is chosen first in its group. However, computing the best-case schedule is NP-hard and a branch and bound method would be too much time consuming. For this reason, we use the lower bound presented in [23, 24] for estimating the optimal value of this criterion.
- *Worst-case schedule*: it provides the worst possible scheduling performance if the selected job is chosen first in its group. This value is computed in polynomial time using the algorithm of [25].
- *Best starting time*: unlike the previous criteria, this one gives direct information about the best possible starting time of jobs. A job can start after the completion time of all its predecessors. This problem is NP-hard and the lower bound presented in [23] is used.

4 Case Study

In this section, we implement the proposed HMI in a real flow shop manufacturing system (Fig. 3). This system, called MPS500 from FESTO Company, is used for constructing short stroke cylinders. It is composed of six workstations, eight pallets and a conveyor transferring pallets (products) from one station to another one.

Fig. 3 MPS500: system under study

4.1 Experimental Protocol

An emulation home-made event-driven program of MPS500 has been built (using JAVA) in order to speed up the manufacturing process and facilitate the experimenting conditions. For each machine, a virtual queue is created in order to store the products. Once the product arrives at the machine, it will be automatically transferred to the queue. If all its predecessors products are already executed on this machine, the product will be then handled by the machine as planned to.

The overall configuration of the setup is presented in Fig. 4. As shown from this figure, a Gantt chart representing the real-time schedule is printed along with the simulation interface. The transportation times on the Gantt chart are considered null.

The input (schedule) of the data module used for the experiment is a flow shop schedule containing 20 products (jobs). We assume that the scheduling decisions can be made only on the fifth workstation, therefore several groups have been constructed on this workstation. There are twelve overall decisions.

The decision-makers of the experiment are 35 students having manufacturing knowledge background and being familiar with MPS500. Despite the fact that

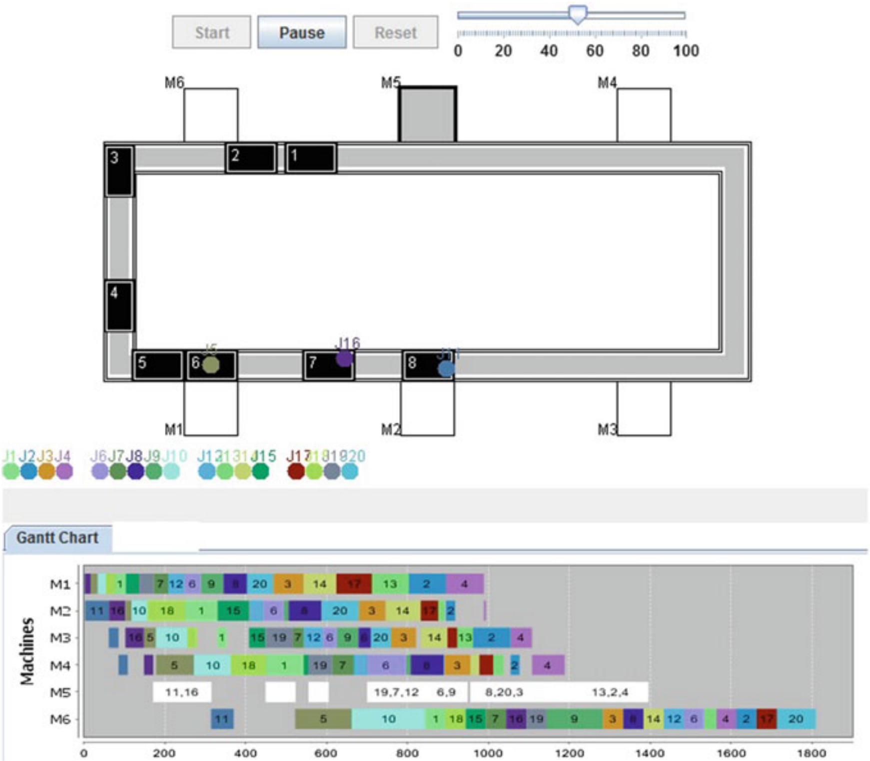


Fig. 4 Simulation program of MPS500

conducting the experiment with students is easier than with professional employees, [5, 26] have shown that, in the field of scheduling, experiments done with students may produce general conclusions applicable to professional situations.

Each time a job of a group arrives at the fifth workstation, it will be stored in the virtual queue of this workstation. Once all jobs belonging to the same group arrive at this queue, the decision-aid interface interrupts the simulation process. The student makes his decision by selecting each time the job that he wants to execute first until the last job of the group. This decision process is repeated until no group of permutable jobs remains.

However, before taking any decision, the student can use the three proposed criteria in order to make his decision. The criteria are computed in the background by the machine module and the transportation time of products is considered null for computing these criteria as well.

To be more realistic, we introduced two perturbations in the schedule. We assume that the first one occurs on decision five and the second on decision eleven. These perturbations are caused by the fact that two jobs took longer time execution than expected. However, the machine module does not consider these perturbations for the computation of the three criteria, and therefore the results may not be correct; it is up to the decision-maker to detect these changes and adapt the results of the decision-aid criteria.

To study the effect of the perturbations on the scheduling performance, we introduced two scenarios:

- In the first scenario, the perturbation can be detected from the HMI interface by checking the planned/executed starting/completion time of jobs. In this case, the student has to calculate by himself the consequences and impact of the perturbation on the current schedule.
- In the second scenario, additionally to the previous one, the perturbations are integrated in the Gantt chart from the simulation interface as shown in Fig. 5. However, the machine module still does not consider these perturbations for the computation of the criteria. Therefore, the student has to verify and correct the results provided by each criterion.

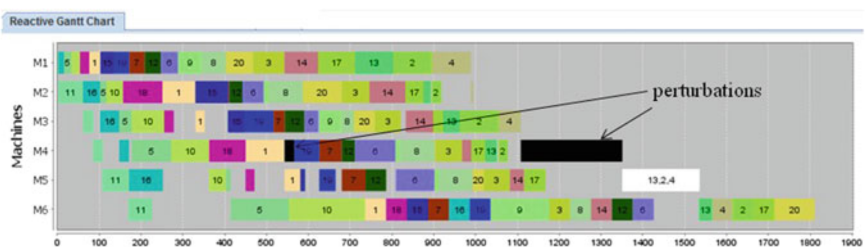


Fig. 5 Representation of the perturbations in the second scenario

The students were divided into two groups. *G1* (of 10 students) and *G2* refer respectively to the first group with the first scenario and the second group with the second scenario. The experiment has been conducted only once for every student.

4.2 Results

The performance of the HMI system has been measured through the makespan objective; once all the decisions are taken, the HMI measures the scheduling performance obtained by each student (referred to as $C_{\max}$). The optimal schedule that can be achieved is referred to as $C^*_{\max}$ and is equal to 1607.

Furthermore, during each decision, the HMI verifies whether the current decision is optimal or not (compared to the schedule represented by $C^*_{\max}$). This measurement is called *local performance* and is represented by a binary variable indicating if a decision is optimal or not. In this way, we can compare the local performance of students before and after the perturbations and its effect on $C_{\max}$.

Table 1 represents the local performance of *G1* and *G2*. For each decision, the percentage of good decisions and the obtained $C_{\max}$ are shown. The index 1 and 2 in the columns' labels refer respectively to *G1* and *G2*. The last column of the table indicates the optimal makespan that can be achieved at the end of each phase regarding the perturbations (the first perturbation occurs at the fifth decision and the second at the eleventh decision).

Table 1 Performance on each decision

Decision number	Local perf. 1 (%)	$C_{\max}1$ (%)	Local perf. 2	$C_{\max}2$	$C^*_{\max}$
1	92	1578	100	1571	1571
2	92	1585	100	1571	1571
3	75	1606	82	1587	1571
4	71	1613	82	1595	1571
5	58	1611	55	1593	1571
6	96	1613	100	1593	1571
7	83	1612	100	1593	1571
8	88	1615	91	1593	1571
9	88	1625	82	1603	1571
10	83	1629	82	1609	1571
11	92	1642	91	1626	1607
12	54	1654	73	1630	1607

It can be shown that *G2* took better decisions than *G1* for almost 67% of the decisions.[2] Furthermore, all students of *G2* were able to take optimal decisions on more than 30% of the decisions. *G1* was able to slightly outperform *G2* on decision 5, 9, 10 and 11. However, the difference between these decisions does not exceed 6%. Overall, $C_{\max}2$ was better than $C_{\max}1$.

The worst decisions of *G1* were taken on decision 5 and decision 12, while the worst decision of *G2* was taken on decision 5. This is due to the perturbations; at the first perturbation, 50% of the students were misled by the decision-aid interface and were not able to correct the values provided by the decision-aid criteria, even though this perturbation did not have a big impact on the planned schedule. This may be due to the inaccuracy of the lower bound used for computing the best-case schedule.

However, after the fifth decision, the students were able to take better decisions. All the students of *G2* were successful in taking optimal decisions during decision 6 and 7. At the last decision, 73% of these students were able to take an optimal decision even though the last perturbation introduced had a significant impact on the planned schedule. In fact, the criteria were supposed to mislead the student by implicitly suggesting the wrong decision as optimal for decision 11 and 12. This was the case for the students of the first group. But overall, the final $C_{\max}$ of both groups is acceptable and represents 2.94% and 1.45% gap distance from $C^*_{\max}$ for *G1* and *G2* respectively.

Furthermore, these results suggest that the representation of the perturbations in a real-time Gantt chart affects the decisions of the human decision-maker, and therefore *G2* was able to have a farther overview about the perturbations impact on the remaining jobs.

Finally, these first results confirmed the usability of the proposed HMI in a practical environment and showed the limits of the control mode, as most of the students were not able to take optimal decisions before the first perturbation. To cope with this drawback, we suggest using different control mode on each decision based on the state of the system. This could significantly improve the local/global-performance of the schedule.

5 Conclusion

In this paper, we proposed a new HMI system for manufacturing scheduling under uncertainties. The aim of the proposed HMI is to improve the cooperation between the human decision-maker and the machine in a context of manufacturing control based on *groups of permutable jobs* method. This method proposes a set of solutions for the scheduling problem, leaving to the human decision-maker the decision of executing one of these solutions in order to cope in real-time with the perturbations in the shop floor.

[2]This result is significant with an error level of 10% using the statistical T-test.

In order to investigate the usefulness of the proposed HMI, we conducted a practical experiment on a real manufacturing system, with students having manufacturing engineering background. The students were divided into two groups, and to each group was affected one scenario. The particularity of the second scenario relies on the perturbations representations on the HMI model in order to help the decision-maker having a farther overview on the perturbation impact and therefore taking better decisions.

The experiments conducted on an emulation program using three decision-aid criteria showed the usability of the proposed HMI. However, the students were not able to take optimal decisions all the time, surprisingly even when the shop floor was stable. In this case, it may be better to switch the control mode to the machine. This research is still in its early stage and should open new questions about the best practice of scheduling under uncertainties using other objective functions.

References

1. Nakamura, N., Salvendy, G.: Design of work and development of personnel in advanced manufacturing. In: Salvendy, G., Karwowski, W. (eds.) Human Planner and Scheduler, pp. 331–354. Wiley, New York (1994)
2. Herroelen, W., Leus, R.: Project scheduling under uncertainty: survey and research potentials. Eur. J. Oper. Res. **165**, 289–306 (2005)
3. Van de Vonder, S., Demeulemeester, E., Herroelen, W.: A classification of predictive-reactive project scheduling procedures. J. Sched. **10**, 195–207 (2007)
4. Hoc, J.M., Mebarki, N., Cegarra, J.: L'assistance a l'opérateur humain pour l'ordonnancement dans les ateliers manufacturiers. Le travail humain **64**, 181–208 (2004)
5. Sanderson, P.M.: The human planning and scheduling role in advanced manufacturing systems: an emerging human factors domain. Hum. Factors **31**, 635–666 (1989)
6. Jackson, S., Wilson, J.R., MacCarthy, B.L.: A new model of scheduling in manufacturing: tasks, roles, and monitoring. Hum. Factors **46**, 533–550 (2004)
7. Crawford, S., MacCarthy, B.L., Wilson, J.R., Vernon, C.: A review of existing knowledge on human factors of planning and scheduling. Cogn. Technol. Work **1**, 63–77 (1999)
8. Trentesaux, D., Pesin, P., Tahon, C.: Distributed artificial intelligence for FMS scheduling, control and design support. J. Intell. Manuf. **11**, 573–589 (2000)
9. Crawford, S., Wiers, VCS: Investigating the work of industrial schedulers through Field Study. IEEE Trans. Softw. Eng.—TSE (2001)
10. Gasser, R., Fischer, K., Wäfler, T.: Decision making in planning and scheduling: a field study of planning behaviour in manufacturing. In: Fransoo, C.J., Waefler, T., Wilson, R.J. (eds.) Behavioral Operations in Planning and Scheduling, pp. 11–30. Springer, Berlin (2011)
11. Berglunda, M., Karltuna, J.: Human, technological and organizational aspects influencing the production scheduling process. Int. J. Prod. Econ. **110**, 160–174 (2007)
12. Wezel, W., Cegarra, J., Hoc, J-M: Allocating functions to human and algorithm in scheduling. In: Fransoo, C.J., Waefler, T., Wilson, R.J. (eds.) Behavioral Operations in Planning and Scheduling, pp. 339–370. Springer, Berlin (2011)
13. Gilovich, T., Griffin, D., Kahneman, D.: Heuristics and Biases: the Psychology of Intuitive Judgment. Cambridge University Press, Cambrighe (2002)
14. Hardman, D., Macchi, L.: Thinking: Psychological Perspectives on Reasoning, Judgment and Decision Making. Wiley, Chichester (2003)
15. Plessner, H., Betsch, C., Betsch, T.: Intuition in judgement and decision making. Taylor and Francis (2008)

16. Hoc, J.M., Guerin, C., Mebarki, N.: The nature of expertise in scheduling: the case of timetabling. Hum. Factors Ergon Manuf. Serv. Ind. **24**, 192–206 (2012)
17. Gacias, B., Cegarra, J., Lopez, P.: Scheduler-oriented algorithms to improve human-machine cooperation in transportation scheduling support systems. Eng. Appl. Artif. Intell. **25**, 801–813 (2012)
18. Dimopoulos, C., Cegarra, J., Papageorgiou, G., Gavriel, G., Chouchourelou, A.: Interdisciplinary design of scheduling decision support systems in small-sized SME environments: the i-DESME framework. J. Sched. **24**, 227–254 (2015)
19. Erschler, R.: An approach for real time scheduling for activities with time and resource constraints. In: Slowinski, R., Weglarz, J. (eds) Advances in Project Scheduling. Elsevier (1989)
20. Roubellat, F., Billaut, J.-C., Villaumié, M.: Ordonnancement d'atelier en temps réel: d'ORABAID à ORDO. Revue d'automatique et de productique appliquées. Revue d'automatique et de productique appliquées. **8**, 683–713 (1995)
21. Lopez, P., Roubellat, F.: Production Scheduling. Wiley, London (2008)
22. Cegarra, J., Hoc, J.-M.: Cognitive styles as an explanation of experts' individual differences: a case study in computer-assisted troubleshooting diagnosis. Int. J. Hum.-Comput. Stud. **64**, 123–136 (2006)
23. Yahouni, Z., Mebarki, N., Sari, Z.: New lower bounds for the best-case schedule in groups of permutable operations. In: 15th IFAC/IEEE/IFIP/IFORS Symposium on Information Control in Manufacturing, vol. 48, pp. 1134–1139 (2015)
24. Artigues, C., Billaut, J.-C., Cherif, A., Mebarki, N., Yahouni, Z.: Robust machine scheduling based on group of permutable jobs. In: Doumpos, M., Zopounidis, C., Grigoroudis, E. (eds.) Robustness Analysis in Decision Aiding, Optimization, and Analytics, International Series in Operations Research and Management Science, pp. 191–220. Springer, Switzerland (2016)
25. Artigues, C., Billaut, J.C., Esswein, C.: Maximization of solution flexibility for robust shop scheduling. Eur. J. Oper. Res. **165**, 314–328 (2005)
26. Cegarra, J., Hoc, J.-M.: A cognitive typology of scheduling situations: a contribution to laboratory and field studies. Theor. Issues Ergon. Sci. **9**, 201–222 (2008)

Using Statistical-Model-Checking-Based Simulation for Evaluating the Robustness of a Production Schedule

Sara Himmiche, Alexis Aubry, Pascale Marangé, Marie Duflot-Kremer and Jean-François Pétin

Abstract Industry 4.0 implies new scheduling problems linked to the optimal using of flexible resources and to mass customisation of products. In this context, first research results show that Discrete Event Systems models and tools are a relevant alternative to the classical approaches for modelling scheduling problems and for solving them. Moreover, the challenges of the Industry 4.0 mean taking into account the uncertainties linked to the mass customisation (volume and mix of the demand) but also to the states of the resources (failures, operation durations, ...). The goal of this paper is to show how it is possible to use the simulation based on statistical model checking for taking into account these uncertainties and for evaluating the robustness of a given schedule.

Keywords Statistical model checking · Production schedule · Robustness Stochastic timed automata

S. Himmiche · A. Aubry (✉) · P. Marangé · J.-F. Pétin
Université de Lorraine, CNRS, CRAN, UMR 7039, Campus Sciences,
BP 70239, 54506 Vandœuvre-lès-Nancy Cedex, France
e-mail: alexis.aubry@univ-lorraine.fr

S. Himmiche
e-mail: sara.himmiche@univ-lorraine.fr

P. Marangé
e-mail: pascale.marange@univ-lorraine.fr

J.-F. Pétin
e-mail: jean-francois.petin@univ-lorraine.fr

M. Duflot-Kremer
Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France
e-mail: marie.duflot-kremer@loria.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_26

1 Introduction

After the mechanisation, started in the middle of the XVIIIth century, the mass production and the electrification, at the end of the XIXth and finally the usage of computers and automation, at the end of XXth, the concept of Industry 4.0 assumes that we are at the beginning of a fourth industrial revolution. This revolution is based on the massive digitalisation and the new Information and Communication Technologies for facilitating the emergence of "smart factories". However, its enforcement highlights new challenges dealing with the management and control of the production systems. In fact, this new paradigm offers, of course, new possibilities: flexibility and agility of production resources, communication capabilities of the resources between them and with their environment, local capability of decision (by the resource and/or by the product). However, the usage of these new capabilities leads to new issues like the mass customisation of the products and needs new control strategies of the production.

These new abilities imply also that the production systems of the future will have as intrinsic characteristics: uncertainties on the demand (volume and mix) or even on their fabrication recipes and a need of optimal using of the resources flexibility. To make it possible, it is necessary to challenge the classical control strategies, which often assume that the predicted demand is certain, that the manufacturing routes are perfectly defined and that resources are always available for computing a centralised production schedule—a global allocation of the production operations to the resources and definition of the starting and completion dates of the operations by satisfying the constraints of the considered system.

In this context, the classical approaches for scheduling, mainly based on operations research (mathematical programming, metaheuristics …) become less efficient because they often consider a stable environment without taking into account the dynamics of the system and its perturbations.

The community of Discrete Event Systems (DES) precisely studies concepts for modelling systems with real time and random aspects, offering an appreciable modelling and analysing power. The methods and languages based on the DES theory became in fact a realistic alternative to classical methods for scheduling the production [1, 2].

The objective of this article is to present an approach based on DES models and tools for evaluating the robustness of a production schedule subject to uncertainties on the operations durations on the machines.

The remaining of the article is organised in five sections. The next section is dedicated to the presentation of the production scheduling problem that we want to address. In the second section, the concept of robustness is developed and a robustness definition is proposed for dealing with perturbations that are modelled by stochastic data. The proposed approach for evaluating the robustness level and the DES models that supports this approach are presented in the third section. An academic example is used in the fourth section for illustrating and discussing the approach. Conclusions and perspectives are given in the last section.

2 Definition of the Problem

2.1 The General Scheduling Problem

Classically, production scheduling consists in (i) allocating the production operations to resources and (ii) sequencing the operations on these resources (defining the order of the operations), also satisfying the constraints defined by the considered production system and optimising a criterion (the total production duration, the number of late operations…). Moreover, when the duration of each operation is considered as perfectly known and static, then it is possible to fix the starting and completion dates of each operation.

The most common criterion is the total duration of the schedule (the makespan, classically denoted as C_{max}). Among the common constraints are the precedence constraints that fix the sequence of operations for each product route.

Regarding its definition, the scheduling problem can be considered as an optimisation problem. Thus, the classical operational research tools (mathematical programming, metaheuristics…) have for a long time been the only tools used for solving these problems.

The classical approach for solving this type of problem is the predictive approach: an *off-line* algorithm—the predictive algorithm—computes an optimal schedule S_I^* for a predicted scenario I (fixing the value of each input parameter of the problem), considered as certain and static, and guarantees a local performance measured by a criterion z and valued by z_I^* on I. The performance z_I^* is only guaranteed for this scenario. However, the production system is naturally submitted to perturbations, and thus, the schedule S_I^* is actually applied to a realised scenario I' that is different from the predicted scenario I. Eventually, the application of the schedule S_I^* to I' can lead to constraints violations, making the schedule no more feasible for I'. Moreover, the real measured performance $z(S_I^*, I')$ can be really far from the predicted one z_I^*.

The limits of the classical approach are clearly highlighted here: it does not guarantee any performance if the realised scenario gets away from the predicted scenario. The need to take into account the perturbations and to propose an approach that is able to compute schedules with good performances despite this perturbations is thus a critical issue.

2.2 Scheduling Flexible Manufacturing Systems

A workshop is defined by a set J of products that have to be processed on a set M of machines (the number of machines is given by card (M) and is denoted as $\mathscr{M}$). Each product j has to follow a production route O_j^J that defines a set of operations to be executed for processing the product j. The execution of the operation o_{jk} (kth operation of the route O_j^J) needs a machine m that must be qualified for this operation, occupying this machine during d_{jkm} time units.

The global set of operations available in the workshop is given by $O^J = \bigcup_{j \in J} O_j^J$ and the total number of operations in the workshop is thus given by card(O^J) and is denoted as $\mathscr{O}$).

There exist different types of classical workshops depending of the possible flows of products: Flow-Shop, Job-shop, Open-shop. According to the type of workshop, some precedence constraints between the operations exist, being defined by the routes O_j^j.

Regarding the criterion z, we will consider only the minimisation of the total duration of the schedule (classically denoted as C_{max}). That means that $z = C_{max}(S, I)$ and assesses the total duration of the schedule S on the scenario I (that fixes the duration of each operation on the machines).

2.3 The Uncertainties

The execution duration of an operation o_{jk} on a machine m for which it is qualified can be rarely known with certainty. A reference duration d_{jkm}^{ref} can often be given: it corresponds to the duration given by the technology engineers. In practice, we will consider that the real execution duration is a stochastic data following a probabilistic distribution with expected value d_{jkm}^{ref}.

On the contrary, the routes O_j^j are supposed to be perfectly known and without variation and the machines are supposed to be without failures.

3 Robustness in Scheduling

Tackling uncertainty in scheduling is not a new problem. We can distinguish two strategies when dealing with perturbations in scheduling that are complementary:

- Trying to take into account the perturbations—proactively—before the production starts; building a robust schedule
- Build an on-line schedule taking into account the real state of the system and the real input parameters

Robust scheduling consists in—according to predefined perturbations (meaning that perturbations are seen here as deterministic uncertainty) and their model—building a schedule that is able to guarantee some performances despite the modelled uncertainties. Performances that must be guaranteed clearly depend on the considered problem. An example, these performances could be a deadline to be respected despite uncertainty on the operations duration.

In the first chapter of [3], the robustness is defined as follows: *a schedule is robust if its performance is rather insensitive to the data uncertainties*. This definition, even if it has the merit to make the consensus, remains poorly precise insofar as it needs to define what means "to be rather sensitive to". That means that some metrics for characterising this sensitiveness are necessary.

In the case where the uncertain data are modelled as random variables (possibly in given intervals), we propose to use the following robustness definition.

Definition 1 A schedule S is conditional-robust face to uncertainties modelled by a vector of random variables I if its probability of satisfying a robustness condition is higher than a given threshold P_{lim}. This can be mathematically formalised by the following condition:

$$Pr(z(S,I) \leq L_\lambda) \geq P_{lim} \tag{1}$$

In this definition, $z(S,I) \leq L_\lambda$ is the robustness condition that is expected to be satisfied despite the uncertainties on I. Typically, L_λ and P_{lim} must be defined by the decision maker who first defines L_λ as the expected performance and then P_{lim} as the minimal desired probability for satisfying the robustness condition. In the context of our problem, as defined previously, $z(S,I) = C_{max}(S,I)$. z assesses the global duration of the schedule S for the scenario $I = \{d_{jkm}\}_{jkm}$ that is a vector of random durations d_{jkm} for the operation o_{jk} on the different machines. Moreover, the necessity is assumed that the global duration of the schedule does not exceed a given deadline $\tilde{d}$ despite these uncertainties. The robustness Definition 1 is then true if the robustness condition above has a probability higher than a given probability P_{lim} when the durations d_{jkm} follow a given probability distribution. This can be expressed by the following condition: $Pr(C_{max}(S,I) \leq \tilde{d}) \geq P_{lim}$.

4 A DES-Based Approach for Evaluating the Robustness

The objective of this section is to show how models based on Stochastic Timed Automata and associated Statistical Model Checking can be used for evaluating the robustness level of a given schedule (Definition 1), and how it is possible to use these models for helping the decision maker to decide between several schedules.

4.1 Approach Overview

The proposed approach includes three steps (see Fig. 1).

- The first step consists in computing a deterministic schedule taking into account the information of the workshop (qualifications of the machines, reference operations durations on the machines, routes of the products) that are here considered as certain and static. The computation of this schedule can be done manually by an expert, or using a classical approach (metaheuristics, Mixed-Integer Linear

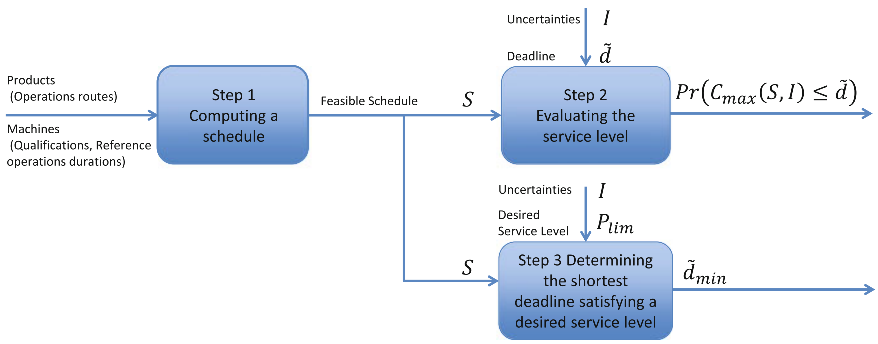


Fig. 1 Procedure for evaluating the robustness of a schedule

Programming ...), or even using DES models and tools as in [2, 4–6]. This step is not addressed in this paper.

- The second step consists, on the schedule obtained after executing step 1, and considering uncertainties on the operations durations, in evaluating the robustness level determined by $Pr(C_{max}(S, I) \leq \tilde{d})$ with $I = \{d_{jkm}\}_{jkm}$ the random vector of operations durations. The deadline $\tilde{d}$ is a reasonable time limit fixed by the decision maker. Its value can be fixed according to a reference duration that can be the makespan of the evaluated schedule on a reference scenario $Iref = \{d^{ref}_{jkm}\}_{jkm}$ denoted by C^{ref}_{max} (we accept to get away from a reference value but without exceeding $X\%$) or can be an arbitrary value corresponding to the horizon of the schedule for instance.
- The third step consists, for the initial schedule of step 1, for the same uncertainties as in step 2, and for a given desired robustness level P_{lim}, in determining the shortest deadline $\tilde{d}_{min}$ that satisfies $Pr(C_{max}(S, I) \leq \tilde{d}_{min}) \geq P_{lim}$.

4.2 Models

In order to implement the procedure presented in the previous section, it is necessary to have models that can represent: the different states of the resources, the operations routes for the products, the allocation and sequencing of the operations according to the given schedule, and the uncertainties on the operations durations. Moreover, it will be necessary to find a way for evaluating the probability to satisfy some properties in order to assess, at the end, the robustness level of the schedule.

There exist several DES tools for satisfying such modelling requirements. We can cite Stochastic Automata [7, 8], Stochastic Petri Nets [9, 10] and Stochastic Automata Networks [11, 12].

4.2.1 Stochastic Timed Automata

Stochastic Timed Automata (STA) [13] are derived from the class of Timed Automata defined in [14]. Informally, a timed automaton as used in UppAal [15] is a model with a finite number of locations, and a finite number of real valued clocks. Discrete and instantaneous transitions can lead from one location to another. Those transitions can be equipped with guards (conditions on clocks that need to be enabled to fire the transition), resets (a subset of clocks whose value will be set to zero on firing the transition) and synchronisation labels (in order to synchronise two automata together). They can also include integer valued variables in guards and updates.

The STA give a probabilistic semantics to TA by probabilistically resolving the deterministic choices in the system, using probabilistic choice between several enabled transitions, and probabilistic distributions (uniform or exponential) to chose the delay before firing the next transition. This model is precisely the one implemented in the statistical model checker UppAal SMC. In the following, we will describe our models using UppAal SMC syntax.

In the remainder of the article, the places will be noted in bold case, the events will noted in italic case, the guards and the invariants will be noted in [] and the updates will be noted in bold case and underlined.

4.2.2 STA Models for a Schedule

In order to evaluate the robustness of a given schedule, it is first needed to model this schedule. Its characteristics are:

- An allocation of each operation to a unique machine,
- The sequencing of the operations on the machines.

Moreover, this schedule has to be admissible, i.e. to satisfy the precedence constraints defined by the routes of the products. In order to model a schedule, two generic models (model patterns) have been defined. According to the solved problem, these patterns are instantiated as many times as necessary.

The first model given in Fig. 2 is an instantiation of the operation pattern and represents the evolution of the states of the operation according to the route of the

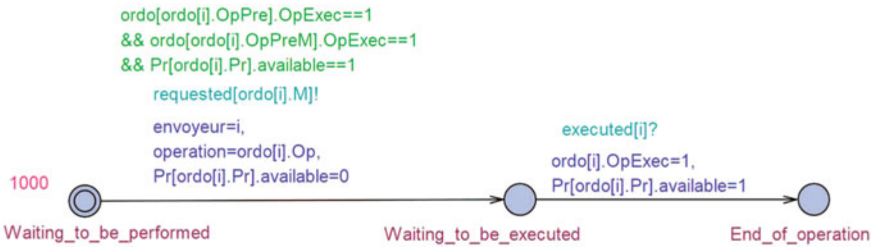


Fig. 2 Instantiation of an operation pattern using UppAal

product to which it is linked, according to the sequencing in the machine allocated to this operation. The operation pattern is based on previous works [5, 16]. We need a copy of this pattern for each operation of each product. The behaviour of the model is as follows.

- In the place ***Waiting_to_be_performed***, the operation is waiting until its enabling conditions (in green) are satisfied. The model can evolve when the previous operation in the route of the product and the previous operation in the sequence on the machine (defined by the schedule) have been completed and if the product is available (not undergoing another operation).
- If this guard is satisfied then the operation sends a request (requested[ordo[i].M]) to the machine allocated, according to the schedule, for executing this operation. Moreover, the status of the product is set to unavailable and the operation now waits to be completely executed in the place ***Waiting_to_be_executed***.
- Upon receiving an event (executed[i]) telling that the machine has completed the operation, the model moves to the final place ***End_of_operation***, and updates variables in order to show that the product is available and the operation has been completed.

The second model given in Fig. 3 is an instantiation of the machine pattern and represents the behaviour of a machine when executing an operation.

The machine pattern has to be instantiated for each machine. It evolves as follows:

- In the place ***idle***, the machine m is waiting for the request requested[m], which means that an operation is requesting to be executed. If the machine is available, it accepts the operation, set its status to occupied, stores the id of the operation, and resets the local clock to compute the execution time.
- The model moves towards the place ***Operation_execution***, and remains there until the operation is completed, which takes a time exponentially distributed which parameter $\lambda = \frac{1}{d_{jkm}^{ref} - d_{jkm}^{min}}$ where d_{jkm}^{min} is the minimal duration of the operation o_{jk} on the machine m.

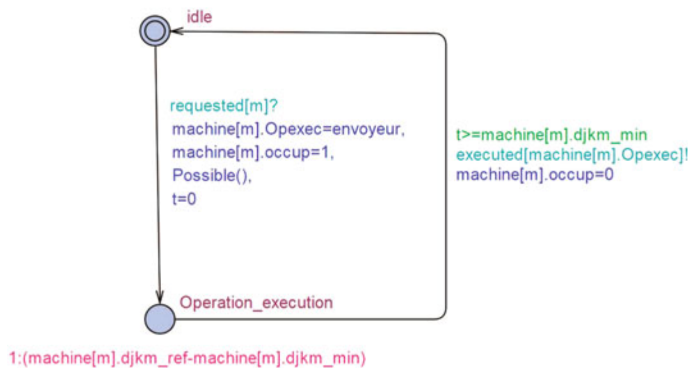


Fig. 3 Model pattern β^m for the machine m

4.3 Evaluation of the Robustness

Based on the previous models and using the model checker UppAal, the steps 2 and 3 of the Fig. 1 are implemented.

Concerning the verification of probabilistic systems, two families of model checkers can be distinguished. The so called (numerical) probabilistic model-checking consists in computing, as precisely as one needs, the exact value of a probability, by using numerical methods based on matrix calculation. The statistical model-checking is based on the sampling of executions in order to estimate a probability by using Monte-Carlo style methods.

The main advantage of the first method is its precision regarding the obtained results (without the risks linked to the usage of statistical methods). However, the second method avoids the combinatory explosion because it does not need to build the complete state space of the system. Moreover, it permits to have richer models (regarding the evolution of the variables or the probability distributions).

In order to answer to the initial issue, i.e. taking into account both time aspects inherent to the scheduling problem and the variability of the system, a tool able to catch deterministic timed activities but also able to introduce some stochastic aspects in other delays (as the treatment of the operations durations) was needed. Regarding these aspects, it was natural to select UppAal SMC [13], a tool that accepts the timed models initially verified by UppAal, but also designed for adding a probabilistic semantics to timed models.

For the robustness level (step 2 of the procedure), the question asked to the model checker is: What is the probability, within deadline $\tilde{d}$ to have completed all the operations?

$$Pr[\leq \tilde{d}](<> forall(i : int[1, NbOp - 1])Ordo[i].OpExec == 1) \quad (2)$$

For step 3, the idea is to find the smallest value d_{min} for which the probability to complete all the operation within d_{min} time units is higher than a fixed value P_{lim}. That means computing the smallest value of $\tilde{d}$ for which the following property is satisfied:

$$Pr[\leq \tilde{d}](<> forall(i : int[1, NbOp - 1])Ordo[i].OpExec) \geq P_{lim} \quad (3)$$

In order to compute this value, a binary search algorithm can be used. The idea is to start with an interval of the possible values of d_{min} and to reduce iteratively this interval until a sufficient precision for the value of d_{min} is reached. This algorithm consists in:

1. Defining a first interval $[d_{inf}; d_{sup}]$ in which the value of d_{min} is guaranteed to be. d_{inf} can be fixed to C_{max}^{ref} and d_{sup} can be fixed to the sum of the durations of all the operations,

2. Reducing the size of the interval $[d_{inf}; d_{sup}]$ by picking the middle value and checking the property (3) with the middle value ($\tilde{d} = \frac{d_{inf}+d_{sup}}{2}$),
3. If the property is satisfied, updating the interval $[d_{inf}; d_{sup}]$ with $[d_{inf}; \frac{d_{inf}+d_{sup}}{2}]$; else updating the interval with $[\frac{d_{inf}+d_{sup}}{2}; d_{sup}]$,
4. Going back to the step 2 of the algorithm with this updated interval (until a given precision is reached).

5 Experiments

The approach, presented in the previous section, has been experimented on an academic example taken from [17] and adapted to take into account the uncertainties on the operations durations. The objective of this example is to schedule eight products. The workshop contains a CNC lathe (M1), a grinding machine (M2), three milling machines (M3 to M5) and two furnaces (M6 and M7). The characteristics of the products to be completed and the workshop are detailed in the Table 1.

Using the method given in [5] and considering a reference operation duration d_{jkm}^{ref} (given in the Table 1), two schedules can be obtained for instance. They are represented in the Figs. 4 and 5. Their respective global durations have the same magnitude. In order to help the decision maker to decide between these two schedules the one to be executed in the workshop, the approach presented in the previous section has been used for each schedule. The robustness level has been first evaluated for each schedule considering that the deadline $\tilde{d}$ has been fixed to 110% of the reference duration fixed by $C_{max}(S, I^{ref})$: 35 for the first schedule and 42 for the second one. That means the decision maker accepts that the duration of the schedule can increase of 10%, but no more.

In order to evaluate the robustness level of the two schedules according to the desired deadline $\tilde{d}$, the following property has been tested on UppAal SMC:

$$Pr[<= 35(resp.42)](<> forall(i : int[1, NbOp - 1])Ordo[i].OpExec == 1) \quad (4)$$

For the first schedule, the obtained probability is 25% and for the second schedule it is 50%. That means that, despite the uncertainties on the operations duration, when executing the second schedule, the products have a probability of 50% to be completed before the desired deadline (42 units of times).

In order to evaluate the smallest deadline that allows a robustness level of 85%, a binary search algorithm has been executing by iteratively testing the following property UppAll SMC:

$$Pr[\leq date\_accept](<> forall(i : int[1, NbOp-1])Op(i).End\_operation\_o_{jk}) \geq 0,85 \quad (5)$$

Table 1 Route of the products and characteristics of the machines

Product	Operation	Resource	d_{jkm}^{min}	d_{jkm}^{ref}
1	$o1\_1$	CNC lathe	4	5
1	$o1\_2$	Milling machine	4	6
1	$o1\_3$	Furnace	1	2
2	$o2\_1$	Milling machine	4	6
2	$o2\_2$	Furnace	1	2
2	$o2\_3$	Milling machine	4	6
2	$o2\_4$	CNC lathe	1	2
2	$o2\_5$	Furnace	5	7
3	$o3\_1$	Milling machine	3	4
3	$o3\_2$	Furnace	8	10
3	$o3\_3$	Milling machine	1	1
4	$o4\_1$	Grinding machine	4	6
4	$o4\_2$	CNC lathe	1	2
4	$o4\_3$	Milling machine	4	6
4	$o4\_4$	Grinding machine	1	2
5	$o5\_1$	Milling machine	4	5
5	$o5\_2$	Grinding machine	4	6
5	$o5\_3$	Furnace	1	2
6	$o6\_1$	Milling machine	2	3
6	$o6\_2$	CNC lathe	2	3
6	$o6\_3$	Furnace	4	6
6	$o6\_4$	Milling machine	1	2
6	$o6\_5$	CNC lathe	3	4
6	$o6\_6$	Grinding machine	3	4
6	$o6\_7$	CNC lathe	5	7
7	$o7\_1$	Milling machine	1	1
7	$o7\_2$	Grinding machine	3	4
7	$o7\_3$	Furnace	1	2
7	$o7\_4$	Milling machine	2	3
7	$o7\_5$	Furnace	6	8
8	$o8\_1$	Milling machine	4	5
8	$o8\_2$	CNC lathe	4	5
8	$o8\_3$	Milling machine	4	6
8	$o8\_4$	Furnace	2	3
8	$o8\_5$	Milling machine	4	6

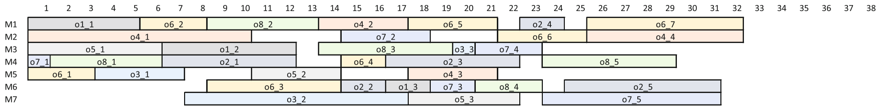


Fig. 4 Possible schedule in 32 time units

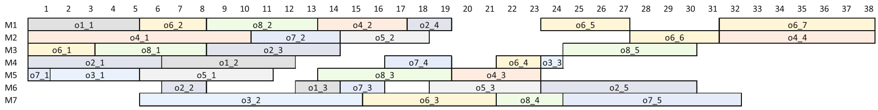


Fig. 5 Possible schedules in 38 time units

For the first schedule, the obtained smallest acceptable deadline is 42 time units and for the second schedule the obtained smallest acceptable deadline is 47 time units. That means that the decision maker has to accept to increase the deadline to 47 time units for obtaining a robustness level of 85%.

Even though the second schedule ends in 38 time units (later than the first one), the first property shows that it is more robust when considering the uncertainties on operations duration. This conclusion is aligned with previous works that show that the most robust schedule is rarely the optimal one (compromise between optimality and robustness).

6 Conclusion

This paper proposes an approach based on Statistical Model-Checking of Stochastic Timed Automata (STA) for evaluating the robustness of one schedule face to uncertainties on the operations durations. This approach first proposes some modelling patterns in STA for modelling a schedule and then proposes to use the Statistical Model-Checking for formally evaluating the robustness level of the modelled schedule. The models that have been proposed are independent of the type of the workshop to be scheduled for. This makes the approach an interesting alternative to classical approaches that are often dedicated to the scheduling problem.

In the future, this approach must be extended and enriched to take into account other types of perturbations (the failures for instance) and other probability distributions. This approach starts with a given schedule to be evaluated. The next step is to propose a method for computing a robust schedule from the scratch according to the modelled perturbations and to a desired robustness level.

References

1. Kobetski, A., Fabian, M.: Time-optimal coordination of flexible manufacturing systems using deterministic finite automata and mixed integer linear programming. J. Discr. Event Dyn. Syst. Theor. Appl. **19**(3), 287–315 (2009)
2. Panek, S., Engell, S., Stursberg, O.: Scheduling and planning with timed automata. In: 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering, pp. 1973–1978. Elsevier (2006)
3. Billaut, J.C., Moukrim, A., Sanlaville, E.: Flexibility and robustness in scheduling. ISTE (2010). https://doi.org/10.1002/9780470611432.ch1
4. Behrmann, G., Brinksma, E., Hendriks, M., Mader, A.: Production scheduling by reachability analysis—a case study. International Parallel and Distributed Processing Symposium (2005)
5. Marangé, P., Aubry, A., Pétin, J.F.: Ordonnancement d'ateliers à partir de patrons de modélisation basés sur des automates communicants. In: 11th International Conference on Modeling, Optimization and Simulation. Montréal, Canada (2016)
6. Subbiah, S., Engell, S.: Short-term scheduling of multi-product batch plants with sequence-dependent changeovers using timed automata models. In: 20th European Symposium on Computer Aided Process Engineering, **28**, pp. 1201–1206. Elsevier (2010)
7. Cassandras, C.G., Lafortune, S.: Introduction to discrete event systems, 2nd edn. Springer (2008)
8. Kwiatkowska, M., Norman, G., Parker, D., Sproston, J.: Performance analysis of probabilistic timed automata using digital clocks. Formal Methods Syst. Des. (2006)
9. Ballarini, P., Djafri, H., Duflot, M., Haddad, S., Pekergin, N.: Petri nets compositional modelling and verification of flexible manufacturing systems. In: Proceedings of the 7th Conference on Automation Science and Engineering (2011)
10. Kritzinger, P., Bause, F.: Stochastic Petri nets—An Introduction to the Theory. Vieweg + Teubner (2002)
11. Plateau, B., Atif, K.: Stochastic automata network of modeling parallel systems. IEEE Trans. Softw. Eng. **17**(10), 1093–1108 (1991)
12. Stewart, W.J., Atif, K., Plateau, B.: The numerical solution of stochastic automata networks. Eur. J. Operat. Res. (1995)
13. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B.: Uppaal SMC tutorial. Int. J. Softw. Tools Technol. Transfer **17**(4), 397–415 (2015)
14. Alur, R., Dill, D.: A theory of timed automata. Theor. Comput. Sci. **126**(2), 183–235 (1994)
15. Behrmann, G., David, A., Larsen, K.: A tutorial on uppaal. In: Proceedings of Formal Methods for the Design of Real-Time Systems, No. 3185 in Lecture Notes in Computer Science, pp. 200–236. Springer (2004)
16. Himmiche, S., Aubry, A., Marangé, P., Pétin, J.F.: Modeling flexible workshops scheduling problems: evaluating a timed automata based approach versus MILP. In: 20th World Congress of the International Federation of Automatic Control. Toulouse, France Accepted (2017)
17. Giard, V.: Gestion de la production et des flux: avec CD livre électronique + Logiciels + Animations. Economica (2003)

Trusted Services for Cyber Manufacturing Systems

Pascal André and Olivier Cardin

Abstract In the context of Industry 4.0, (cyber) manufacturing systems enter in a world of services which is a convenient paradigm to match virtual and physical systems including Cloud computing, big data, Internet-of-Things (IoT) and mobility. The coordination and control of such complex system of actors or components requires methods and techniques to design, verify and deploy the services, possibly on the fly, making service engineering an unavoidable approach to develop new generations of cyber manufacturing systems. In this position paper, we advocate a service based component model that would be helpful to contribute to this goal.

Keywords Manufacturing systems · Service · Contract · Verification · MDE

1 Introduction

The Web revolution enabled connecting not only people but also systems through unified communication protocols. A new step, called Industry 4.0, interleaves computation processes and physical processes in the recent so-called field cyber-physical systems (CPS) which defines a new and multidisciplinary approach that covers many engineering areas such as mechatronics (from electrical or mechanical component), robotics … [1]. This new step arises with new converging technological computer science advances like massively distributed and cloud computing, Internet-of-Things (IoT), revisited artificial intelligence and big data, mobility with variety of computer systems and sensors, new user practices and mobile applications. We face a new interconnected world the production systems are not only connected to the

P. André
LUNAM Université, Université de Nantes, LS2N UMR CNRS, 6004 2,
rue de la Houssinière, 44322 Nantes Cedex, France
e-mail: pascal.andre@univ-nantes.fr

O. Cardin (✉)
LUNAM Université, IUT de Nantes – Université de Nantes, LS2N UMR CNRS,
6004 2 avenue du Prof. Jean Rouxel, 44475 Carquefou Cedex, France
e-mail: olivier.cardin@univ-nantes.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_27

management systems (ERP) but also opened to the client (B2C) and the suppliers (B2B) including supply management, manufacturing, client relationship … Cyber-Physical Production Systems (CPPSs) consist of autonomous and cooperative elements and subsystems that are connected based on the context within and across all levels of production, from processes through machines up to production and logistics networks [2]. Business processes can be connected to manufacturing processes. Cyber manufacturing is a transformative concept that involves the translation of data from interconnected systems into predictive and prescriptive operations to achieve resilient behaviour [3].

The unifying paradigm between processes and systems is the concept of **service**, which current hot topic is the cloud stack XaaS and the Service Oriented Architectures (SOA). Services are also compatible with (high level) business processes in enterprise architecture frameworks. Of course the definition of service varies a lot from one context to another. At low levels a service can be a script, a procedure, an operation or a method, depending on the implementation language. At intermediate levels a service is a process that coordinates other service (through service composition and orchestration) to achieve a specific or independent goal. At high level a service can be interpreted as a business process that monopolizes shared resources (and maybe people) to fulfil all or part of a business or production process. The coordination and control of such complex systems by actors or components require methods and techniques to design, verify and deploy the services, possibly on the fly, making service engineering an unavoidable approach to develop new generation cyber-manufacturing systems. This is part of CPS's requirements mentioned by Wang et al. in the category 'design methodology': "*Research challenges include development of techniques for efficiently integrating or relating multiple models/viewpoints/data sets, CPS design methodology for trustworthy end-to-end services including adaptive and autonomous systems, and platforms for safe and secure CPS design that underpin design methodology, facilitating integration and establishing desired system level properties.*" [1]. As mentioned by Bauer et al. the traditional automation pyramid is dissolving and manufacturing IT is moving towards service-orientation and app-orientation [4]. As an example, in cloud manufacturing, SOA was identified to meet the requirements of all higher level manufacturing CPS layers due to the reduced time constraints present [5].

Service engineering is still a craft activity, usually at the implementation level [6, 7]. Two main levers are still required to go further. First, we need service models that can fit to various semantics and various granularity levels. Indeed, the concept of cyber-physical production system (CPPS) covers many classes of (physical) systems, from the manufacturing workshop to the power distribution network. Taking the example of Holonic Manufacturing Systems, the control of such systems is often recursive, if not fractal, in order to aggregate the available resources and enable heterarchic control architecture. Therefore, services that might be used at various levels of the architecture need to fit various granularity levels and the portability of services between different applications with their own semantic requires an adaptability of the services to be effective. Second, we need analysis tools to check the service model properties on various aspects (structure, dynamics, functional and non-functional), and model transformations to compute new models or to generate code in the sense of *model driven engineering (MDE)*.

In this position paper, we advocate a service based component model with embedded contracts that would be helpful to reach this goal. This model is abstract, in the sense of Service Oriented Architectures (SOA), to capture various semantics. Applying the proposed methodology with early verification from formal models reinforces confidence in the services early in the development process: they are corrected, they embed evaluated test data, and contracts can be reinforced. This helps correcting software as soon as possible in the process and allows applying then advanced development techniques such as agile ones (thanks to the qualified test data we constructed) or Design-by-Contract (thanks to the contract we reinforced).

We present the general process in Sect. 2, the service models in Sect. 3, the service contract in Sect. 4 and the way to supply them the service implementation of model transformation in Sect. 5. Section 6 illustrates the approach on a small part of a vehicule control system. Related works are discussed in Sect. 7. In conclusion, we draw perspectives for manufacturing.

2 A Development Process for Trusted Services

We assume a general presentation, which is not specialized to manufacturing systems, where services denote *software services*, whatever implementation links them to physical systems. Service based systems, as well as component based systems, are realized according to a composition principle (with several composition operators); high level services are hierarchically composed of lower level services where the low level services (the leafs) are atomic. We assume that (software) *components* are containers of related services; the reason for grouping services in a component can be multiple: same provider, same business, same time, same job, same deployment node …

The design activities associated to composition are: (1) to define the goal and properties (functional and non-functional), (2) to find the adequate services, (3) to define a service orchestration and (4) to check whether the composition is correct and consistent with the service goal and properties and delivers a trusted composite service. Steps (2) and (3) can be manual or automatic, depending on the repository management. The implementation activities are transformations of design models to the target code. In order to build trusted services we need to specify formal models, then to verify them before implementation and storage on the shelf. This development sketch is illustrated by Fig. 1.

The principle of component or *service on the shelf* is still not accessible in practice: software designers cannot pick up services and compose them the way one can design electronic devices, mechanical assemblies or building constructions. Furthermore automating these activities remains challenging. The service market is hardly available at design time and is often achieved at the implementation level where the service API is poorly defined by a signature and a comment. Consequently

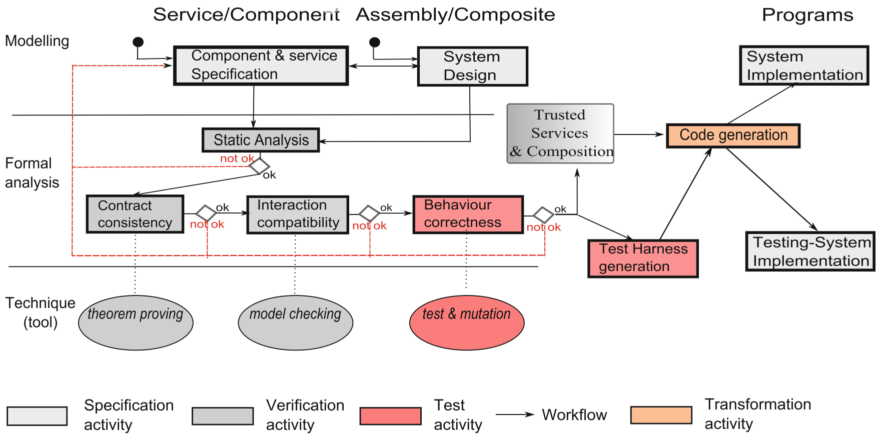


Fig. 1 Applied specification, verification and implementation process

searching candidate service requires human actions and understanding. We present Sect. 3 a service model with rich interfaces to express contracts. In practice, services are merely final services, considered as a whole. This means that service requirements are already resolved and we cannot change the providers of an intermediate service except if for service owners. In the service model of Sect. 3, the requirements are specified by service contracts as detailed in Sect. 4. Last, there is a gap between the service paradigm, well-defined at the architecture level, and its implementation, where the notion of service does not exist: a service is implemented by XML descriptions and programming statements. If service (or component) oriented programming would exist, the traceability links would be more explicit. In Sect. 5 we set the basis for service implementation that preserves the service structure and traceability.

3 Service Modelling

In Service-based Component (SbC) models, a functionality is implemented by services provided by components. Provided services are not necessarily atomic calls and may have a complex behaviour, in which other services might be needed (called). These needs are either satisfied internally by other services of the same component, or specified as required services in the component's interface. The required services can then be bound to provided services from other components, which might also require others, and so on. A provided service needs all its direct and indirect dependencies satisfied in order to be available for use. The process of providing trusted service on the shelf or designing is based on a triple: service model, property model and verification model.

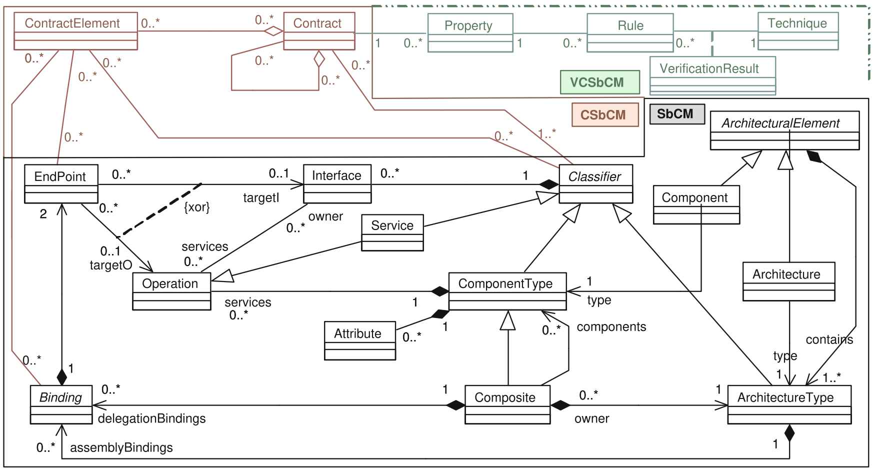


Fig. 2 Abstract service based component metamodel, contracts and verifications

In the service based component model *SbCM* of Fig. 2, the components (types) are characterised by an abstract state and services. Components are assembled on their access points (*EndPoint*) which can be ports, interfaces or services according to the concrete languages (SCA, Sofa, Fractal, Kmelia). A service specification may vary from the simple signature to the detailed description of dynamic behaviour with communications and service composition. An assembly (also called architecture) binds components, possibly using other assemblies, upon a clientship relation (classical use of client/server contract). The connector paradigm is reduced to *binding* because it is usually the representation in component implementations. A composite that encapsulates assemblies, and thereafter components, upon an inclusion relation (parent/child contract) that may promote child observable features (state, services). The involved features cover structural, functional and dynamic aspects. The service behaviour and the communications are described, depending on the specification language, by control structures, regular expressions, process algebra, state machines.... Detailed information enables service verification.

4 Service Contracts and Verification

We separate the specification and verification concerns. As illustrated by Fig. 2, the *SbCM* meta-model is enriched with a contract layer (*CSbCM*) and a verification layer (*VCSbCM*). Layering enables to disconnect system models from properties to check (the contracts) and tools to verify. The layers enable to target verification on specific verification tools. The contract layer is used to establish a service relation between *classifiers* (e.g. services, components), A classifier is a kind of black-box encapsulating an implementation. A contract is more than the assertions of Design by contract when including the composition of services, service interactions, quality of service [8]. According to [9], *a Trusted Component* is a reusable software element possessing specified and guaranteed property qualities. The notion of contract is helpful to model various kind of correctness properties. But it should be made precise and extended to cope with the expressiveness of the SbC models. The properties, such as *interoperability*, are classified at different contract levels:

1. **Static**. Does a service or a component give enough information about its interface(s) in order to be (re)usable by others? The service call should respect the service signature (names and types). The signature matching between the involved services of component interfaces covers at least name resolution, visibility rules, typing and subtyping rules.
2. **Architectural**. Are the required services available all along the dependency chain? Are the assembly links corrects? Assuming that services can be composed from other (sub)services, connecting services is possible only if their structures are compatible (but not necessary identical).
3. **Functional or computational**. Do the services do what they must do? These correctness properties may be checked both on individual service in regard to their container component and on the compositions. This third level deals with *service compliance*. If the services use a Hoare-like specification, post-conditions relate to their pre-conditions. The caller pre-condition is stronger than the called one. The called post-condition is stronger than the caller's one. Each part involved in the assembly should fulfil its counterpart of the contract.
4. **Behavioural or interactions**. Do the involved services interact correctly in a collaboration? The *behavioural consistency* property states that the execution of the service actions does not lead to inconsistent states (such as deadlock). The properties depends on the interaction model features: sequential/concurrent, call/synchronisations, synchronous/asynchronous, pair/multipart communication, shared data, atomic/structured actions …
5. **Quality of service**. Does a service fulfil non-functional requirements (time, size, cost …)? For example, several services can be candidate but some are more efficient, more secure or more available than others.

In order to cope with different meanings and different contexts, we introduce the notion of **multi-level contract** in [8] where a contract is defined at different structure levels (service, component, assembly, composition) according to different expected requirement levels (syntax compatibility like CORBA IDL, structural compatibility, functional compatibility, behavioural compatibility and QoS compatibility). This cross vision of contracts provides a convenient framework to master both the incremental construction of SbC and the verification of multi-aspect properties by combined techniques. Also it provides a foundation for searching services in libraries (services and components on the shelf).

The verification layer is based on a triple <*property, rule, technique*> (*cf*. Fig. 2). There is no wide-spectrum formal languages that enable the verification of all the properties in once. Usually, a modelling language is coupled with a verification technique and dedicated tools. e.g. the B notation is supported by the Atelier-B or Rodin theorem provers, LOTOS is supported by the CADP model checker, etc. To target various types of properties, a full formal analysis requires *model transformations* to target the adequate tools for the kind of properties to check. For example, in the case of the process of Fig. 1, the functional contract can be checked using the theorem prover; the interaction contract is verified by model checking and the conformance of the behaviour against the contract is controlled using testing techniques. Remark that model testing can use the same MDE facilities than the one of service implementation (see Sect. 5) to generate code for test harnesses.

5 Service Implementation

Model Driven Engineering (MDE) emphasizes the use of models and meta-models to improve the software productivity and some aspects of the software quality such as maintainability or interoperability. According to the principles of MDE [10] one can refine service models to executable programs by plunging them in technical domains (frameworks) and preserving the service structure and traceability.

Figure 3 illustrates the mapping between a platform independent model and a platform specific level. This can result from a sequence of transformation, not necessary from a one step transformation. For example, one can target a specific implementation framework model (e.g. REST or SOAP web service model, CPS control languages …) but high level models can also be intermediate models (e.g. WSDL or BPEL business processes, agent cyber model …). This depends on the technical architecture and also on the available model transformations. Some primitive functions are kept abstract at the model level and have to be mapped to concrete ones.

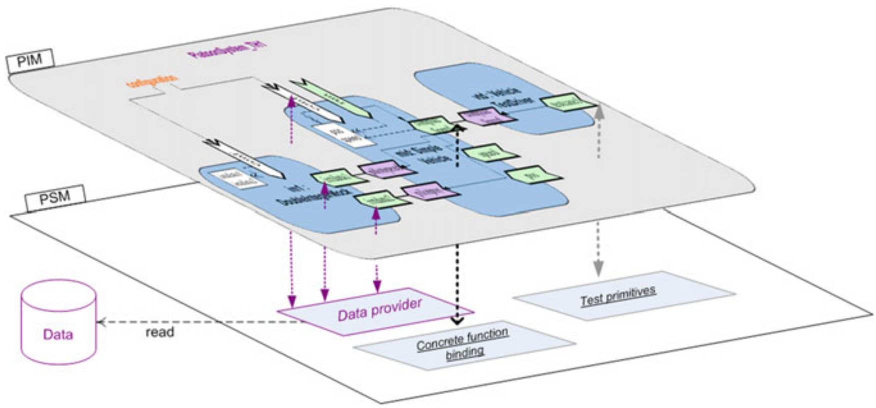


Fig. 3 Model concrete data and function mapping

This verification step checks whether these mappings are well-defined and consistent or not. The links with physical CPS devices can be obtained similarly.

6 Case Study

This section will briefly introduce the application context, an illustrative example and will include various references on previous works and experimentations.

We experimented the approach with a simplified version of a *platoon of vehicles* case study, which can share similarities with AGV (Automated Guided Vehicle), where the embedded application should ensure safety properties such as avoiding collisions or not losing a vehicle. The vehicles and the driver are components which interact to know their position and speed in order to control their move. We consider here only the speed and the position (X axis only) of the vehicles. The vehicles are designed to follow their predecessor (which they consider to be their pilot) except the first one which follows a component taking the role of the driver. The driver is assumed to be a special kind of vehicle that controls its own values according to a target position. Each running vehicle can compute its own speed by considering its current speed and position, its predecessor's position and speed and a safety distance with respect its predecessor.

The approach has been experimented *only at the software layer* with the COSTO, a CASE tool dedicated to the development of Service based Component (SbC) Systems. The specification language is Kmelia, a wide-spectrum language dedicated to the development of correct SbC models [11]. To be short Kmelia allows defining service behaviours with extended state machines using an action language which includes synchronous communication on channels. Components contain several services and several components can be assembled using client-server links. An assembly link establishes an (explicit) communication channel. A composite component encapsulate assemblies in which services of the subcomponents can be promoted at the composite level.

Figure 4 shows a small architecture composed of a *driver* and three *vehicle* components. We use the SCA notation [12] to make explicit the component's interfaces with provided and requires services (called references in SCA). Each component has a configuration service conf (used when instantiating the component), and a main service run, which is launched automatically after configuration (autorun). The run and conf services assign values to the vehicle's state. The run service activates the vehicle behaviour and services to give their position and speed; it's a loop that ends when the platoon reach its goal. The computeSpeed service reads the vehicle's state to compute the next speed. Other services like stop which interrupts a vehicle, have been omitted for simplicity.

In Kmelia the components, assemblies and compositions can be analysed according to various facets and levels, as detailed in [11]. The general verification approach is explained in [8]. It is instrumented by the COSTO tool, a set of Eclipse plugins including an editor, a type checker, several analysis tools and exportations to

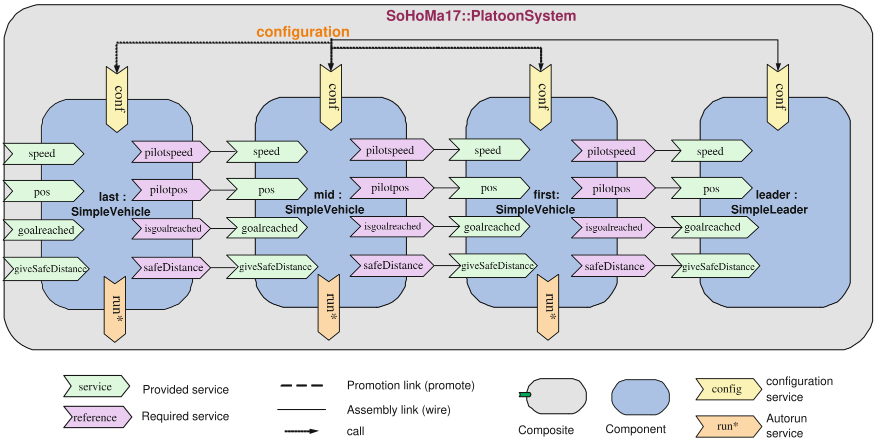


Fig. 4 Component model of the platoon system

external tools like MEC, CADP (Lotos), AtelierB and Rodin (Event-B), and also a Java code generator and an assistant to generate and execute test harnesses [13].

Previous works detail the verification techniques: the verification of functional contracts inside services/components [11] and in service composition [14] (theorem proving), the verification of behavioural compatibility [15] (model checking), the verification of consistency between service contracts and concrete behaviour [16] (model testing). For sake of space we redirect the reader to these references and we just enumerate in the remaining some kind of properties that could be checked on the platoon example (S-safety, L-liveness, F-Failure …):

S1 A vehicle moves in the bad direction because the given values are false or incorrect (not in the scope, not the same unit or representation…). This can lead to collision with other vehicles, people or environment.

S2 A vehicle leaves the platoon to be autonomous (with no driver) but does not reconnect its follower to its driver (dynamic service reconfiguration).

S3 A hacker intercepts the communications between two vehicles to send non-authorized information; the service contract should not be interruptible (broken subservice dependency).

L1 A vehicle waits because it cannot find its predecessor position or speed (service availability) or understand the given values (bad parameters or result type) or cannot read the values on time (bad QoS or synchronization).

L2 A vehicle cannot move because the physical situation is not consistent with the virtual situation (twin failure), for the same reasons as above.

L3 The platoon is stopped due to bad configuration or reconfiguration; e.g. the vehicle is (directly or indirectly) driven by its own position and speed (infinite loop).

F1 An intermediate vehicle accelerates or decelerates suddenly due to a bad interpretation of its received values; the followers will have the same jerky behaviour.

F2 The platoon turns round due to bad general configuration or reconfiguration; at least this is the case L3 where the position and speed are given with a delta of time which differs from case F1.

7 Related References

This section points out relevant information from related references

Bauer et al. [4] discuss current trends in manufacturing IT. They provide a good overview on manufacturing services and also apps, including concepts and implementation. Independent service vendors (ISV) are able to offer their services on the platform and users should be able to orchestrate services according to their needs in order to flexibly adapt to changing market conditions. We bring a new perspective on it by providing service engineering to put this need in practice.

As far as services can encapsulate or abstract not only software but also physical behaviours (like digital twins do), orchestration refers to smart adaptable assembly systems as introduced by ElMaraghy [17] which creates the possibility of cross-fertilization of ideas. We provide a deep semantic information that will complete the abstract layer of service interoperability ontologies [18].

Our proposal can take place in the integration part of the general frame for design, modelling, simulation and integration of cyber physical systems given by Hehenberger et al. [19] which includes mechatronic and Internet of Things (IoT). Abstraction and interoperability are key concepts for such systems. They denote often a lack of clearly specified and documented interactions and interfaces between the various disciplines and involved components and hence mutual understanding in communication is hindered. Our model can help in providing an abstract service layer for trust interoperability like an Architecture Description Language(ADL) which is more service oriented than the ADLs they mentioned. This takes place in the Service Enablement Layer of Monostori et al. [2] or in the 3rd level of the 5 C architecture for implementation of Cyber-Physical System of Lee et al. [20]. Services can play roles in both *twin* models and the integration to higher level layers including production monitoring and also business processes or manufacturing simulation as in [21].

Liu et al. introduce a new paradigm of Cyber-Physical Manufacturing Cloud (CPMC) to bridge gaps among cloud computing, cyberphysical systems, and manufacturing [22]. They propose a four-layer service-oriented CPMC architecture where what we call atomic services could belong to layer 2 (resource virtualization) and our composed service would belong to layer 4 (core cloud) that handles API, security and publication. We think that our model of rich interfaces could be integrated in layer 4 with verification facilities. Morgan and O'Donnell investigated whether SOA could be integrated into a cyber-physical manufacturing execution system to enable cloud monitoring [5]. SOA was identified to not meet the deterministic requirements of low levels but meets those of all higher level manufacturing CPS layers due to the reduced time constraints present.

Our service meta-model covers a broad kind of services; it can be enriched with specialized meta-model like the UML4IoT to exploit IoT in cyber-physical manufacturing systems [23].

Compared with the work of Gamboa Quintanilla et al [17], our model is less specialized but complementary. Their concept of service corresponds to our notion of atomic service while their processes are atomic service assemblies. Building high level services would be possible at implementation only. We could extend this model with required services, communications between services, dynamic behaviour... These concepts make possible rich interfaces and powerful automatic service search and composition.

8 Conclusion

The service orientation, enforced by cloud computing, becomes dominant not only in software engineering but also in cyber physical systems, production systems and manufacturing control. The principles of component- and service-based computing have been set since two decades but the practice is far from these theories and the "on the shelf" principle: services are usually composed by programmers. The Kmelia model improves the service selection (search the adequate service) and the service orchestration (check the deep compatibility before assembling). Some features of Kmelia are really innovative e.g. rich interface and required service enables to really have modular descriptions of services which are fundamental for service interoperability or service substitution.

Of course, the illustrated example is not directly related to CPS and manufacturing but we are convinced that providing tool assistance to statically detect potential errors when assembling services would be helpful in that case. Open perspectives are: (1) to extend existing CPS manufacturing service models to rich interfaces to enable searching and orchestration facilities, (2) to check statically or on the fly the provision of services, and (3) to target various implementation model, (SOA, WSDL ...) or to generate implementation to deploy.

References

1. Wang, L., Törngren, M., Onori, M.: Current status and advancement of cyber-physical systems in manufacturing. J. Manuf. Syst. **37**, 517–527 (2015)
2. Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., Ueda, K.: Cyber-physical systems in manufacturing. CIRP Ann. Manuf. Technol. **65**(2), 621–641 (2016)
3. Lee, J., Bagheri, B., Jin, C.: Introduction to cyber manufacturing. Manuf. Lett. **8**, 11–15 (2016)
4. Bauer, D., Stock, D., Bauernhansl, T.: Movement towards service-orientation and app-orientation in manufacturing IT. In: 10th CIRP Conference on Intelligent Computation in Manufacturing Engineering–CIRP ICME '16. (Ed.) Roberto Teti: Doriana M. D'Addona Procedia CIRP vol. 62, pp. 199–204 (2017)

5. Morgan, J., O'Donnell, G.E.: The cyber physical implementation of cloud manufacturing monitoring systems. In: 9th CIRP Conference on Intelligent Computation in Manufacturing Engineering—CIRP ICME '14 Procedia CIRP vol. 33, pp. 29–34 (2015)
6. Morariu, C., Morariu, O., Borangiu, T.: Customer order management in service oriented holonic manufacturing. Comput. Ind. **64**(8), 1061–1072 (2013)
7. Rodrigues, N., Leitão, P., Oliveira, E.C.: Self-interested service-oriented agents based on trust and QoS for dynamic reconfiguration. In: Borangiu, T., Thomas, A., Trentesaux, D. (eds.) Service Orientation in Holonic and Multi-agent Manufacturing, Studies in Computational Intelligence, vol. 594, pp. 209–218. Springer (2015)
8. André, P., Attiogbé, C., Mottu, J.M.: Combining techniques to verify service-based components. In: Proceedings of the International Workshop on domAin Specific Model-based Approaches to verification and validation, AMARETTO@MODELSWARD 2017, Porto, Portugal, February 19–21, 2017, pp. 645–656 (2017)
9. Meyer, B.: The grand challenge of trusted components. In: ICSE '03: Proceedings of the 25th International Conference on Software Engineering, pp. 660–667. IEEE Computer Society, Washington, DC, USA (2003)
10. Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice, 1st edn. Morgan & Claypool Publishers (2012)
11. André, P., Ardourel, G., Attiogbé, C., Lanoix, A.: Using assertions to enhance the correctness of kmelia components and their assemblies. Electr. Notes Theor. Comput. Sci. **263**, 5–30 (2010)
12. OSOA: Service component architecture (sca): Sca assembly model v1.00 specifications. Specification Version 1.0, Open SOA Collaboration (2007)
13. André, P., Mottu, J.M., Sunyé, G.: Costotest: a tool for building and running test harness for service-based component models (demo). In: Proceedings of the 25th International Symposium on Software Testing and Analysis, ISSTA 2016, pp. 437–440. ACM, New York, NY, USA (2016)
14. André, P., Ardourel, G., Messabihi, M.: Component service promotion: contracts, mechanisms and safety. In: Barbosa, L.S., Lumpe, M. (eds.) 7th International Workshop on Formal Aspects of Component Software, FACS 2010. Lecture Notes in Computer Science, vol. 6921, pp. 145–162. Springer (2010)
15. Attiogbé, C., André, P., Ardourel, G.: Checking Component Composability. In: 5th International Symposium on Software Composition, SC'06. LNCS, vol. 4089. Springer (2006)
16. André, P., Mottu, J.M., Ardourel, G.: Building test harness from service-based component models. In: Proceedings of the Workshop MoDeVVa (Models 2013), Miami, USA, pp. 11–20 (2013)
17. Quintanilla, F.G., Cardin, O., L'Anton, A., Castagna, P.: A modeling framework for manufacturing services in service-oriented holonic manufacturing systems. Eng. Appl. Artif. Intell. **55**, 26–36 (2016)
18. Bianchini, D., De Antonellis, V.: Ontology-based Semantic Interoperability Tools for Service Dynamic Discovery, pp. 323–333. Springer, London (2006)
19. Hehenberger, P., Vogel-Heuser, B., Bradley, D., Eynard, B., Tomiyama, T., Achiche, S.: Design, modelling, simulation and integration of cyber physical systems: methods and applications. Comput. Ind. **82**, 273–289 (2016)
20. Lee, J., Bagheri, B., Kao, H.A.: A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manuf. Lett. **3**, 18–23 (2015)
21. Tao, F., Cheng, J., Cheng, Y., Gu, S., Zheng, T., Yang, H.: Sdmsim: a manufacturing service supply–demand matching simulator under cloud environment. Robot. Comput.-Integr. Manuf. **45**, 34–46 (2017). special Issue on Ubiquitous Manufacturing (UbiM)
22. Liu, X.F., Shahriar, M.R., Sunny, S.N.A., Leu, M.C., Hu, L.: Cyber-physical manufacturing cloud: architecture, virtualization, communication, and testbed. J. Manuf. Syst. **43, Part 2**, 352–364 (2017). High Performance Computing and Data Analytics for Cyber Manufacturing
23. Thramboulidis, K., Christoulakis, F.: Uml4iot–a uml-based approach to exploit iot in cyber-physical manufacturing systems. Comput. Ind. **82**, 259–272 (2016)

Local Search with Discrete Event Simulation for the Job Shop Scheduling Problem

Hugo Zupan, Janez Žerovnik and Niko Herakovič

Abstract Multi-start local search heuristics Remove and Reinsert that is based on a simple schedule constructing heuristics is tested on several benchmark instances of the job shop scheduling problem. The heuristics provides very good near optimal solutions within reasonably short computation time. The implementation within a plant simulation software is compared to the build-in genetic algorithm.

Keywords Job shop scheduling problem · Remove and reinsert heuristics
Genetic algorithm · Discrete event simulation · Digital twin

1 Introduction

Optimization of assembly and handling systems and processes (AHSS) is important in terms of reducing costs, shortening lead times, delivery terms, etc., thus ensuring the competitiveness of enterprises. It has been clearly shown [1] that disturbances can cause reduction of the overall effectiveness of equipment (OEE), which can represent up to 50% of the costs. For this reason, it is essential to optimize AHSS. Different approaches and methods have been used to optimize AHSS in order to effectively achieve the optimum, respectively nearly optimal solution [2–4]. One of the most famous optimization problems of AHSS is the Job-Shop Scheduling Problem (JSSP).

H. Zupan (✉) · J. Žerovnik · N. Herakovič
Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia
e-mail: hugo.zupan@fs.uni-lj.si

J. Žerovnik
e-mail: janez.zerovnik@fs.uni-lj.si

N. Herakovič
e-mail: niko.herakovic@fs.uni-lj.si

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_28

Over the past few decades, a great number of studies have been made on job-shop scheduling problem (JSSP). JSSP can be regarded as a scheduling problem and is one of the most challenging combinatorial optimization problems. It is of both theoretical and practical interest, being highly popular in production industry. For conventional JSSP, it is usually assumed that all time parameters are known exactly and in deterministic values. An instance of JSSP can be described as follows: we have a set of n jobs that need to be operated on a set of m machines. Each job has its own processing route; that is, jobs visit machines in different orders. Each job may need to be performed only on a fraction of m machines, not all of them. The task is to determine a processing order of all jobs on each machine that minimizes the total flow time.

Another usual assumption is that each job can be processed by at most one machine at a time and each machine can process at most one job at a time. When the process of an operation starts, it cannot be interrupted before the completion; that is, the jobs are non-preemptive. The jobs are independent; that is, there are no precedence constraints among the jobs and they can be operated in any sequence. The jobs are available for their process at time 0. There is unlimited buffer between machines for semi-finished jobs, meaning that if a job needs a machine that is occupied, it waits indefinitely until it becomes available. There is no machine breakdown (i.e. machines are continuously available) [5].

Setup times of machines are typically sequence dependent (or SDST), that is, the magnitude of setup strongly depends on both current and immediately processed jobs on a given machine. For example, this may occur in a painting operation, where different initial paint colours require different levels of cleaning when being followed by other paint colours. We also assume that setup is non-anticipatory, meaning that the setup can only begin as soon as the job and the machine are both available [5].

The JSSP is known to be an NP-hard optimization problem [6]. Therefore, application of metaheuristics for the JSSP is justified when looking for optimal or near optimal solutions in reasonable amount of time. This paper proposes such an algorithm, based on Remove and Reinsert algorithm (or RaR).

It is well known [7] that using discrete event simulation or digital twin is very effective tool for “what-if” scenarios, for every type of production system. In our case we have transformed real production system with all the features and limitations into virtual factory. The idea is that the metaheuristic proposes initial and iteratively improved schedules of orders while the discrete event simulation performs “what-if” scenario for each proposed schedule thus providing the quality measure of the schedule. This process is repeated until the metaheuristic can no longer provide better schedule.

The rest of this short contribution is organized as follows. In the next section, the metaheuristics RaR is outlined and its operation is illustrated with an example and results in Sect. 3. Concluding remarks are formulated in Sect. 4.

2 Motivation

Our algorithm is inspired by some applications of several similar heuristics that appeared under various names. These heuristics were successfully applied to the probabilistic travelling salesman problem (PTSP) [8] and the asymmetric traveling salesman problem [15]. It may be rather surprising that such a simple heuristics outperforms much more complicated metaheuristics such as (in this study) a commercial implementation of a genetic algorithm. We think that this phenomenon is not that unexpected, see [9] and the references there. In other words, as the basic idea of the heuristics is very simple and somehow natural for the particular problem, the authors speculate that this may in fact be a reason for good results.

RaR can be regarded as a local search based on RaR neighbourhood or as a constructive heuristics. For example, on the travelling salesman problems (TSP, ATSP, PTSP) a new neighbour of a given solution is obtained by first removing a number of cities from the tour, and then reinserting them one by one into the best position that does not change the relative order of the other cities. The tour constructing heuristics on TSP (ATSP, PTSP) starts with a small subset of cities, computes their optimal permutation, and then inserts the other cities in arbitrary order. An iteration of iterative improvement consists of first removing some of the cities and then reinserting them in arbitrary order. The results on PTSP [8] and ATSP [15] were encouraging where they were competitive with the best known heuristics of the same type. At the time, this was rather surprising as TSP is one of the most extensively studied problems in combinatorial optimization and operational research.

For the Job-shop Scheduling Problem (JSSP) that is studied here we apply the basic ideas above as follows. Below we first explain the basic neighbourhood which corresponds to a perturbation of a feasible solution into a new feasible solution. Given a parameter *k* and a feasible solution, *k* jobs are removed and reinserted, or, in this case we better say, the relative positions in the sequence of the selected *k* jobs may change. In contrast to RaR on the travelling salesman problems, we do not remove all the jobs at the same time, but remove and reinsert the selected jobs one at a time. In a pseudo programming language, it could be written as

```
Procedure GenerateNeighbourBasic(S0,k) Returns(S1)
1.  S1=S0
2.  Choose k jobs J_k
3.  For i=1 to k do
    a.  Select a job j_i from J_k (J_k = J_k -j_i)
    b.  Insert j_i into S1 on the best position
4.  Return(S1)
```

Considering the basic neighbourhood given by procedure GenerateNeighbourBasic, several other neighbourhoods can be naturally defined. Here we first define a neighbourhood called large neighbourhood defined by procedure GenerateNeighbourLarge that can change the given solution substantially. The GenerateNeighbourLarge procedure first removes a subset of jobs, but, before reinserting them, it solves the sub-problem to optimality. This implies that large *k* have to be used as otherwise the procedure would be very time consuming.

```
Procedure GenerateNeighbourLarge(S0,k) Returns(S1)
1.  S1=S0
2.  Choose k jobs J_k
3.  Remove jobs J_k from S1
4.  Find optimal order of jobs J-J_k within S1
5.  For i=1 to k do
    a.  Select a job j_i from J_k (J_k = J_k -j_i)
    b.  Insert j_i into S1 on the best position
6.  Return(S1)
```

Local search using different neighbourhoods is not a new idea. For example, it is extensively studied under name variable neighbourhood search [10, 11].

Note that GenerateNeighbourLarge can also be seen as procedure for generating an initial solution. Given any *S0*, GenerateNeighbour-Large (*S0*, *k*) is a heuristics that provides a good solution. (Note that there is some clear analogy to the well-known Arbitrary Insertion tour constructing heuristics for TSP.)

A version of GenerateNeighbourLarge that we use below allows selection of the *k* jobs to be perturbed outside the procedure. For this aim we define the jobs to be selected by their positions, therefore a different name for this variant:

```
Procedure GenerateNeighbourLargePos(S0,P_k) Returns(S1)
1.  S1=S0
2.  Let J_k be k jobs at positions P_k
3.  Find optimal order of jobs J-J_k within S1
4.  For i=1 to k do
    a.  Select a job j_i at the next position among P_k
    b.  Insert j_i into S1 on the first best position
5.  Return(S1)
```

In our implementation of RaR, we run a multi-start RaR-based local search heuristics. However, as we avoid randomization after the initial solution is given, the sequence of selected neighbours is predefined. In more details, the heuristics used in the experiment is:

```
Generate a Random initial solution S
While (there is time left) do
    Repeat
      S0=S
         Run a sequence of moves:
         For w=1 to n-m
            P = positions w+m,…,n
            S1 = GenerateNeighbourLargePos(S,P)
          S = better between S and S1
    Until S not better than S0
```

We conclude this section with a list of remarks emphasizing some basic facts regarding our implementation of the heuristic.

1. We run a multi-start of iterative improvements of the large neighbourhood GenerateNeighbourLargePos.
2. We speed up the implementation by avoiding randomization in the first experiments. In particular, the choices of the jobs that generate a sub-problem are always a sequence of $(n-k)$ jobs starting at some position, say w. Again, this seemingly counterintuitive decision is based on the speculation that such sub-problems may provide relatively good starting solution before reinsertion.
3. Removing jobs from their set in the computed solution is performed one at a time, thus all the jobs contribute to the cost of intermediate solutions. This decision was taken because we have observed that completely removing many jobs may cause that the properties of the sub-problems differ too much from the full problem and, consequently, even very good solutions of a sub-problem may provide poor starting solution before reinsertion.
4. The testing environment is the software program Siemens Tecnomatix Plant Simulation, which is used for evaluations of our scenarios and for comparison of our RaR algorithm with built-in genetic algorithms. Therefore the most natural measure of time here is the number of scenarios. We also measure wall clock time, but note that this information is not very useful, as the software provides a user friendly front end with lots of graphics etc. that we do not control but tends to be very time consuming.

3 Results

For computational results the RaR algorithm was combined with discrete event simulation.

As mentioned before, using discrete event simulation or digital twin is a very effective tool for “what-if” scenarios, for every type of production system [12]. In our case we have transformed JSSP with all the features and limitations into digital

Fig. 1 Basic concept of the digital twin

twin. The idea is that the metaheuristic proposes initial and iteratively improved schedules of orders while the discrete event simulation performs "what-if" scenario for each proposed schedule, thus providing the quality measure of the schedule. This process is repeated until the metaheuristic can no longer provide better schedule (Fig. 1).

Recall that we assume that all queues are FIFO. Hence the algorithm optimizes only the schedule on the source.

The algorithm was tested on some well-known and one of the most used benchmark instances for the JSSP—LA01 to LA05 problems from Lawrence [13] and MT06, MT10 and MT20 problems from Fisher and Thompson [14]. We compared the RaR algorithm with the built-in "Siemens" genetic algorithm (SGA), which is already installed in the Siemens programming environment Plant Simulation [12]. The results are presented in Table 1. Note that both algorithms always started with the same initial scheduling *O1*, *O2*, …, *On*.

From the results we see that RaR algorithm finds a very good solution in a relatively short time compared to SGA. The great advantage of RaR algorithm is that it is not necessary to store large amounts of data, since the algorithm works sequentially, and in almost every step takes only the best solution while discarding the others.

We made also a test to see the effect of different initial schedules on our RaR algorithm, to see how they effect on end results. The results showed that with different initial scheduling different best solutions are possible. The results were tested on benchmark LA02 and are presented in Table 2. Note that in 7 runs, the quality of the best solution by RaR has improved. In one of the runs it was even better (754) than the solution given by SGA (758).

As mentioned in Sect. 2, in real production there are setup times of machines which are sequence dependent—this problem is called Sequence Depended Setup Times JSSP (SDST JSSP). To the best of our knowledge, there are no benchmark instances in the literature; we plan to generate a dataset of several instances for a more extensive experiment, results to be reported in the full paper.

We have a production process with 9 work places where operations are carried out. The plan is to produce 10 orders. For each order, the technology procedure and

Table 1 Comparison of results between SGA and RaR

Problem	SGA best solution	SGA time (s)	RaR best solution	RaR time (s)
LA01	705	13	705	3
LA02	758	17	778	3
LA03	679	21	681	2
LA04	660	10	660	2
LA05	593	6	593	2
MT06	59	5	59	1
MT10	1092	24	1092	4
MT20	1496	114	1496	9

Table 2 Effect of different initial scheduling on best solution found by RaR algorithm

Initial scheduling	RaR best solution
O1, O2, O3, O4, O5, O6, O7, O8, O9, O10	778
O9, O8, O7, O10, O5, O3, O6, O1, O4, O2	758
O5, O8, O6, O3, O1, O7, O2, O4, O10, O9	766
O5, O8, O6, O3, O1, O9, O2, O4, O7, O10	**754**
O5, O9, O7, O10, O8, O3, O6, O1, O4, O2	758
O8, O6, O5, O9, O10, O7, O2, O1, O3, O4	778
O10, O9, O8, O7, O6, O5, O4, O3, O2, O1	778

the sequence of operations are known (see Table 3). Raw material goes from material storage and the finished products are stored in the products storage. The order is always moving from machine no. one towards the no. machine nine. Orders can be transported between machines only one at a time. If there are several orders at the same operation, then the orders are awaiting a free operation in the buffer of needs located in front of the operation.

Table 3 The matrix of the type and sequence of operations

	Machine (*i*)								
Order (*j*)	Mach. 1	Mach. 2	Mach. 3	Mach. 4	Mach. 5	Mach. 6	Mach. 7	Mach. 8	Mach. 9
N1	X	X	X	X	X				X
N2	X				X	X	X	X	X
N3	X				X	X	X	X	X
N4	X	X	X	X	X				X
N5	X				X	X	X	X	X
N6	X				X				X
N7	X				X				X
N8	X	X	X	X	X				X
N9	X				X				X
N10	X	X	X	X	X	X	X	X	X

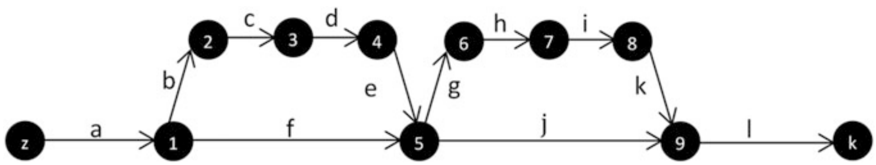


Fig. 2 Weighted directed graph of production process

Table 4 The results of GA and RaR algorithm on SDST JSSP

Algorithm	Total time	… for finding best solution	Quality of the best solution
Instance with 10 orders			
GA	1 min	33 s	2468
RaR	20 s	8 s	2468
Instance with 100 orders			
GA	4 h 5 min 22 s	4 h 5 min 22 s	14385
RaR	29 min 30 s	22 min 41 s	13768

For each operation there is information about the setup time for the machine and operational time for the order. The setup time is sequence-dependent on both current and immediately processed order.

From the layout of production the weighted directed graph was made (see Fig. 2), nodes represents machines and weight ($k = a, b, \ldots, l$) represents transport time between machines.

The RaR algorithm was also tested for two SDST JSSP problems with 10 jobs and with 100 jobs. In both cases we compared the algorithm with the genetic algorithm. The results are presented in Table 4.

The characteristics of the genetic algorithm (GA) are:

- Instance with 10 jobs: 50 generations: size of generation = 100;
- Instance with 100 jobs: 500 generations; size of generation = 100.

According to the results of the first tests that have been carried out (reported above and some of these tests that are not described here), we can say that the RaR algorithm works very well and in quick time gives good solutions.

4 Conclusions

This paper proposes Remove and Reinsert heuristics for the job-shop scheduling problem. Preliminary results show that it provides very good solutions thus nearly minimizing the expected average flow time in a reasonable amount of calculation time.

For executions of “what-if” scenarios of the initial schedules, the discrete event simulation software—Tecnomatix Plant Simulation was used. A comparison of RaR algorithm with the Genetic Algorithm which is built-in module in Tecnomatix Plant Simulation software showed that RaR algorithm finds good solution in shorter time compared to Genetic Algorithm. We have to mention that we did not change any parameters of the built-in GA. Maybe some better tuning of parameters for GA would improve its performance, but this was not possible as the software we use does not allow such user intervention.

Motivated by the promising results outlined here, we have also executed “what-if” scenarios for different priority rules inside the production processes. The results showed that by changing priority rules of processing the orders on the machines, we get even shorter flow time of all orders. Details will be given in the full paper.

In our future work, further experiments will be conducted to shorten the calculation time of getting the optimal solutions from the RaR algorithm.

Finally, as the RaR heuristics performs remarkably well on the JSSP, it may be worth considering the same idea on the other versions of the JSSP and some other NP-hard problems.

Acknowledgements This research work has been funded by the GOSTOP program, contract no. C3330-16-529000, co-financed by Slovenia and the EU under ERDF. The second author was supported in part by ARRS, the Research Agency of Slovenia.

References

1. Ylipää, T.: Correction, prevention and elimination of production disturbances. PROPER project description, Department of Product and Production Development (PPD), Chalmers University of Technology, Gothenhurg (2002)
2. Debevec, M., Simic, M., Herakovic, N.: Virtual factory as an advanced approach for production process optimization. Int. J. Simul. Model. **13**(1), 66–78 (2014)
3. Herakovic, N., Metlikovic, P., Debevec, M.: Motivational lean game to support decision between push and pull production strategy. Int. J. Simul. Model. **13**(4), 433–446 (2014)
4. Pindeo, M.L.: Scheduling: Theory, Algorithms, and Systems. Springer New York Dordrecht Heidelberg London, Springer Science + Business Media, LLC (2012)
5. Naderi, B., Fatemi Ghomi, S.M.T., Aminnayeri, M.: A high performing metaheuristic for job shop scheduling with sequence-dependent setup times. Appl. Soft Comput. **10**, 703–710 (2010)
6. Ombuki, B., Ventresca, M.: Local search genetic algorithms for the job shop scheduling problem. Appl. Intell. **21**, 99–109 (2004)
7. Eskandari, H., Rahaee, M.A., Memarpour, M., Hasannayebi, E., Malek, S.A.: Evaluation of different berthing scenarios in Shahid Rajaee container terminal using discrete-event simulation. In: Simulation Conference (WSC), Winter (2013)
8. Zerovnik, J.: A heuristics for the probabilistic traveling salesman problem. In: Rupnik, V. Bogataj, M. (eds.) Proceedings of the International Symposium on Operational Research (SOR’95), pp. 165–172, Slovenian Society Informatika, Ljubljana (1995)
9. Zerovnik, J.: Heuristics for NP-hard optimization problem: simpler is better! Logistic Sustain. Transp. **6**(1), 1–10 (2015)

10. Mladenović, N., Hansen, P. and Brimberg, J.: Sequential clustering with radius and split criteria. Cent. Eur. J. Oper. Res. 21 (Supplement-1): 95–115 (2013)
11. Hansen, P., Mladenović, N.: Variable neighbourhood search methods. In: Encyclopedia of Optimization, pp. 3975–3989 (2009)
12. Tecnomatix Plant Simulation, Siemens PLM Software. http://www.emplant.de/english/fact%20sheet%20plant%20simulation.pdf. Accessed on 26 June 2017
13. Lawrence, S.: Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (Supplement). Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania (1984)
14. Fisher, H. Thompson, G.L.: Probabilistic learning combinations of local job-shop scheduling rules. In: Muth, J.F., Thompson, G.L. (eds.) Industrial Scheduling, pp. 225–251. Prentice Hall, Englewood Cliffs, New Jersey (1963)
15 Brest, J., Zerovnik, J.: An approximation algorithm for the asymmetric traveling salesman problem. Ric. Operativa **28**, 59–67 (1999)

A Negotiation Scenario Using an Agent-Based Modelling Approach to Deal with Dynamic Scheduling

Tsegay Tesfay Mezgebe, Hind Bril El Haouzi, Guillaume Demesure, Rémi Pannequin and André Thomas

Abstract In order to ensure an overall efficient production performance, the industrial experiences and research activities have demonstrated the interest of hybrid control systems, which couple a predictive scheduling with a distributed reactive control. In this context, it is commonly accepted that using the multi-agent systems (MAS) paradigm enhances the reactivity to treat disturbances and improves the decision making process of a shop floor. Each agent can have different capabilities (evolution, learning etc.) and the whole system, based on the agent interaction, leads emerging behaviours to dynamically adapt the production schedule. This paper is aimed to develop and simulate a negotiation scenario to deal with disturbed manufacturing processes. The scenario was implemented on the basis of TRACILOGIS test-bed platform. The negotiation protocol consists in setting the best sequential priority based on some performance indicators. This protocol is compared with a purely reactive production mode.

Keywords Multi-agent systems · Negotiation protocol · Distributed reactive control

T. T. Mezgebe (✉) · H. B. El Haouzi · G. Demesure · R. Pannequin · A. Thomas
Université de Lorraine, CRAN, UMR 7039, Campus Sciences, BP 70239, 54506 Vandœuvre-lès-Nancy cedex, France
e-mail: tsegay-tespay.mezgebe@univ-lorraine.fr

H. B. El Haouzi
e-mail: hind.el-haouzi@univ-lorraine.fr

G. Demesure
e-mail: guillaume.demesure@univ-lorraine.fr

R. Pannequin
e-mail: remi.pannequin@univ-lorraine.fr

A. Thomas
e-mail: andre.thomas@univ-lorraine.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_29

1 Introduction

Nowadays, manufacturing industries are facing many challenges to deal with the increase of the variability on all supply chain processes [1] and to propose a well-designed production process to execute a requested product at the right time to customers in accordance with the specified schedule and timing. As part of this, these industries are continuously seeking for new control systems that assure reactivity to unpredictable events. In order to ensure reactivity to unpredicted events and provide an overall efficient production performance, the industrial experiences and research activities have demonstrated the interest of hybrid control system that couple a predictive centralized mode with a distributed reactive mode based on the reactivity of Cyber Physical System (CPS).

For instance, when schedules become rapidly unusable because of frequent changes, using dynamic scheduling methods to face such disturbances has become crucial. To realize this surmount, intelligent optimization protocols and algorithms are used to define agents within the working environment that must continuously negotiate to reach at converged and optimal offer.

In this paper, a negotiation scenario among product and resource agents cooperating to set best sequential priority-based production processes is presented. MATLAB simulation based on the TRACILOGIS test-bed platform is used to compare reactive and negotiating production heuristics. The rest of the paper is organized as follows. Section 2 elaborates the justification and cites related works. After this, Sect. 3 demonstrates the negotiation heuristic comparing it with the pure reactive one. Section 4 analyses the comparative results of reactive production and negotiating production processes. Finally, Sect. 5 concludes and formulates remark mainly relating to the role of agent consensus for optimizing the manufacturing system of a shop floor.

2 Background and Related Works

Previous studies have revealed that multi-agent based control systems treat changes and disturbances observed within a shop floor. An agent is an autonomous entity that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators [2, 3]. Naturally, agents in general, but also acting in non-deterministic environments must be prepared for the possibility of failure. Due to this non-deterministic nature, Wooldridge [3] has raised two questions that could always come when one needs to implement cooperative agents:

- The *design*: how can one build agents that are capable of independence and autonomy to successfully carry out their tasks?
- The *sociability*: how can one build agents that are capable of cooperating with other agents?

In answering these queries, an agent has to coordinate with other agents in several different ways. A system that consists of a group of agents that can potentially collaborate with each other is a multi-agent system (MAS) with the capability to perceive, reason, communicate in order to solve in common problems that are beyond individual competences [3, 4].

In MAS, the simple presence of multiple agents makes the environment to appear dynamic from the point of view of each agent, within the control system they are part of—typically distributed control [5]. As a result, to create an environment that provides an infrastructure allowing the communication and interaction of agents based on protocols, Weiss [6] has defined the knowledgeability, predictability, reactivity, and sociability of agents as basic characteristics of multi-agent environment. To fulfil these characteristics, agent cooperation and negotiation deserves special attention. For instance, Dimopoulos and Moraitis [7] state that individual agents can generate and execute their plans independently; however, as they operate in the same environment, conflicts may arise and as a result they need to coordinate their course of action in order to avoid harmful interactions. Zambrano et al. [8] and Wooldridge [3] have also demonstrated that negotiation among agents aim at providing robust-predictive-reactive scheduling and also to tackling myopia. A chain reaction based on product's perception could minimize the impact of disturbances and allow intelligent products to take into account other products' sequences. With this, a certain estimation of system efficiency and performance could be possibly made, providing a certain degree of predictability of the system's behaviour.

Tonino et al. [9] investigate different agent negotiation approaches including game-theoretic methods [10], heuristic-based approaches [11], and argumentation-based approaches [12]. They emphasize the importance of exchanging information and explanations between negotiating agents in order to mutually influence their behaviours. Moreover, they propose a formal protocol handling negotiation dialogues between many agents ($n \geq 2$); the protocol is supposed to be run as long as offers are not analysed.

3 Agent Negotiation Based on TRACILOGIS Test-Bed Platform

If an unexpected event having significant impact on a master production schedule has occurred, negotiating rescheduling and control is expected to save the master schedule right after the caused interruption. To detect such an event, different intelligent solutions have been used considered in previous studies even though some challenges are still delaying the manufacturing execution system of a shop floor production process. Previous research has outlined three challenges;

evaluating future system performances and designing efficient switching mechanisms are the first two challenges approached. The concern at this stage is how to design an efficient mechanism leading to the best decision accepted by all control agents in order to determine the whole system to remain globally optimized.

The scenario proposed in our research is simpler; to demonstrate it we consider three product types P_1, P_2, and P_3. For each of them ten intelligent products are launched in the TRACILOGIS test-bed platform. The physical system is composed by two physical resources (M_1 and M_2) with their cyber parts (resource agents and product agents). Each product agent tries to follow a standard routing sheet shown in Fig. 1, based on the run time given in Table 1.

In the routing sheet shown in Table 1 and Fig. 1, we consider that P_1 arrives at M_1 for its first processing operation (O_1), then moves to M_2 for its second processing task (O_2), and returns them to M_1 for its third processing (O_3) with its completion time C_{ij} (where 'i' is the ith product agent and 'j' is the jth resource agent) and then completes its route. If the product routes without optimized

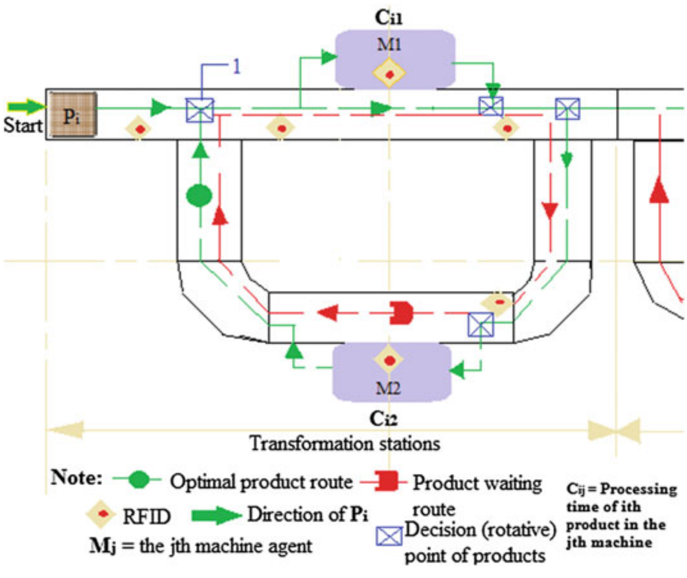


Fig. 1 Media input of the TRACILOGIS platform

Table 1 Run time and due date of three product types

Products	Run time on each machine			Due date	Remark
P_1	$M_1(3)$	$M_2(1)$	$M_1(4)$	10	Completes its route
P_2	$M_1(11)$	$M_2(3)$	–	18	Completes its route
P_3	$M_1(8)$	–	–	7	Completes its route

operational sequence, this will lead to a higher setup time. Hence, the route has been established following reactive and negotiating heuristics using MATLAB simulation run in the media input of the TRACILOGIS platform, in order to provide the best priority based sequential production process.

3.1 The Proposed Negotiation Model Based on Critical Ratio

In the negotiation heuristic, all agents are set up to cooperate between them by computing and analyzing the products' needs and the resources' capacity. The need (or "intention") of each product agent '*i*' is to arrive and trigger the process at each resource agent '*m*'; each product broadcasts its own intention to each resource that it is approaching based on its starting time, run time and finish time, as expressed in Eq. (1).

$$\mathrm{v}_{\mathrm{i(m)}} = [\mathrm{a_i} \ \mathrm{r_i} \ \mathrm{f_i}] \tag{1}$$

where $\mathrm{v}_{\mathrm{i(m)}}$ represents the intention of product '*i*' for each resource '*m*', and $\mathrm{a_i}$, $\mathrm{r_i}$, and $\mathrm{f_i}$, are respectively the starting time, run time and finish time of product '*i*' relative to each resource '*m*'. It results therefore from Table 1:

$$\begin{aligned} &v_{1(1)} = [0\ 3\ 3],\ v_{1(2)} = [3\ 1\ 4],\ \text{and}\ v_{1(3)} = [4\ 4\ 8] \\ &v_{2(1)} = [3\ 11\ 14],\ v_{2(2)} = [14\ 3\ 17] \\ &v_{3(1)} = [14\ 8\ 22] \end{aligned}$$

Each product '*i*' computes its intention and sends it to all other products and to the resource so that its intention is understood by all agents and executable on the resource. After each product sends its intention, resources are expected to be used in order to utilize their available run time (i.e., to reduce the setup time and to support the competitive priority of cost) based on the utilization model given in Eq. (2):

$$\text{Utilization} = \frac{\text{Average processing time}}{\text{Maximum processing capacity}} \tag{2}$$

To achieve this usability, resource agents select products based on their critical ratio order with an intention to process first that product having the smallest critical ratio. Hence, each resource must calculate the critical ratio (CRi) of product 'i' based on the model:

$$CR_i = \frac{\text{Due date} - \text{current date}}{\text{Total shop time remaining}} \tag{3}$$

where the remaining total shop time includes the setup, processing, routing, and expected queuing times of all remaining operations, including the operation being scheduled.

Finally, depending on the objective of maximizing the resource's run time, each resource gives priority to the product that best meets the due date, as shown in Table 1. Consequently, after a product '*i*' recalculates its intention and sends it to all products, the other products evaluate the intention set by product '*i*' and accept it if it doesn't affect their predetermined critical ratio; otherwise, they ask product '*i*' to revise its intention. For instance, when P1 and P3 meet in decision point number '1' of Fig. 1 while P1 is routing to M1 for its third operation and P3 begins its route, P3's intention is to precede P1. Hence, P3 will query P1 to wait wherever it is so that P3 will first process in M1. Otherwise, P3 will route a loop even though its critical ratio is less than the critical ratio of P1 in M1. Likewise, while products are prioritizing themselves based on their intention, they have to be validated by resources as products' intentions may lead to idleness of resources, which in turn increases the setup time. Thus, resources recalculate their setup time at every intention of products and review the critical ratio based on product priority.

4 Simulation Experiment Results

As stated earlier, making several simulation runs for critical ratio-based negotiation helps to converge to a better optimization. In this case, fifty MATLAB simulation runs (the results are presented in Appendix) have been executed for each production heuristic by considering the resource utilization rate, product lateness (the tardiness), and the global makespan.

Initially, the reactive production was simulated taking into account the "change in product intention" principle such that agents simply react to what happened in their routes. When products do not allow sending their correct intention, resources become inefficient and are susceptible to high setup time when the products are tardy; as a result, the global makespan increases linearly. Once the result of this reactive production is obtained, the routing sheet is shifted to negotiating production heuristic where "update the intention and routing" approach is used. Accordingly, Figs. 2, 3, and 4 compare and analyse these two production approaches and show that the pure reactive production mode is almost linear, while negotiation is nearly alternating to try satisfying the intentions of product agents.

Figure 2 compares the simulation results considering the lateness of products. In the negotiating production mode, agents are capable to reduce the lateness to a

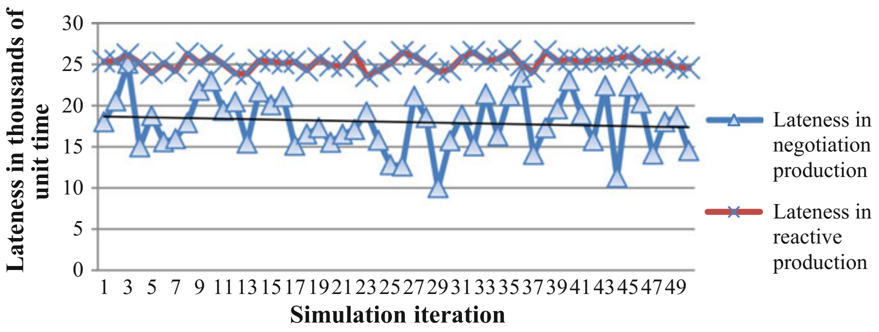


Fig. 2 Product lateness of the two production processes

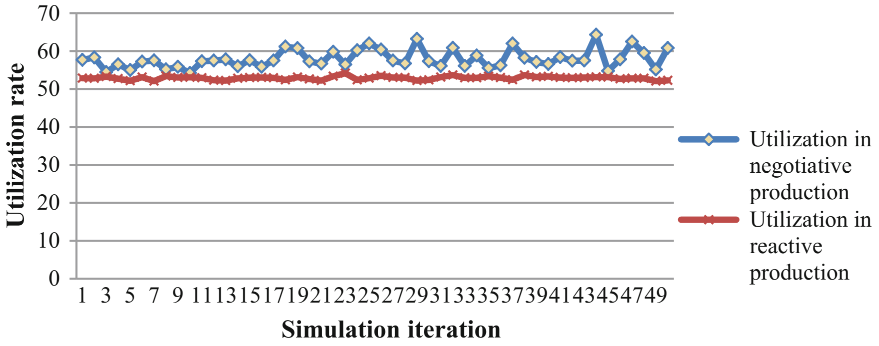


Fig. 3 The utilization rate analysis

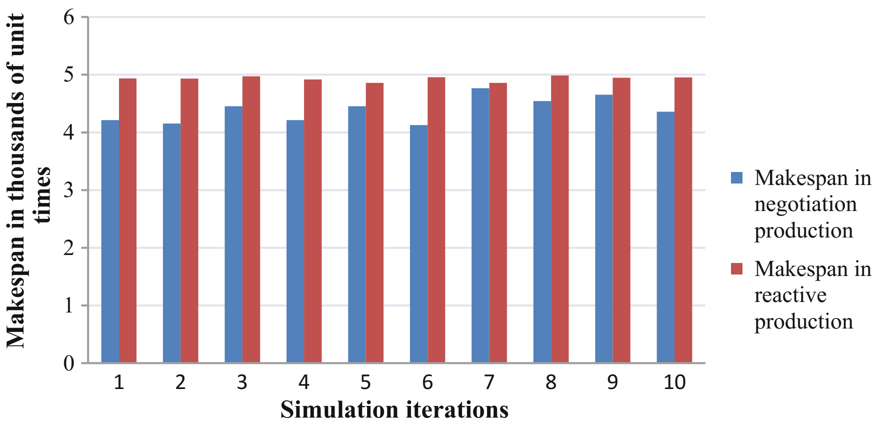


Fig. 4 The makespan analysis

minimum of 10,000 unit times in one of their simulation runs, but in reactive production their minimum lateness has resulted to be 23,600 unit times. This shows that in the former approach, products arrive at resources 13,600 unit times earlier than in the second production heuristic. Meanwhile, the product lateness in the routing sheet has reduced, on average, by 28.67% as a result of the environment created for negotiation rather than for reactive mode. Similarly, the utilization rate of resource (M_1) is better than in the reactive approach while the routing sheet follows negotiating production, see Fig. 3. On average, when the routing sheet follows the negotiation approach, it has utilized the resource (M_1) 9.51% better than in the reactive production approach.

Considering the global makespan, Fig. 4 clearly shows that in the reactive approach, once agents have set their sequential route, they almost continue to follow this route rather than setting another optimal route that helps minimizing their makespan. But in the case of negotiating production, agents update their pre-set route to minimize their makespan at least by 1.89% of the reactive approach.

Finally, these simulation experimental results indicate that the role of intelligent multi-agents system is to initiate and support the decision making process of manufacturing processes. To achieve this goal, agents must continuously negotiate among themselves and reach a final consensus in order to converge to a common offer.

This consensus is reaching in dynamic systems an agreement regarding the common interest depending on the states of all agents [13]. However, even though many related research works concentrate on this subject, there are currently important consensus challenges to impose a common goal to each entity/agent that has its own priorities, goals and levels of understanding its environment.

5 Conclusion and Future Works

This paper has described a negotiation scenario to simulate the product's routing sheet based on the media input of the TRACILOGIS test-bed platform and to compare the result with reactive sub-scenarios. The simulated results show that negotiation among resource and product agents bring sensible improvement with respect to reactive approaches. For example, the product lateness in the routing sheet has been reduced, on average, by 28.67% as a result of the environment created for negotiation rather than for reactive mode. Similarly, the utilization rate of resources (M_1) while the routing sheet uses negotiation heuristic is 9.51% better than the reactive production approach. These values demonstrate that the negotiation heuristic has brought sizable improvement in the routing sheet considering the resource utilization, product tardiness, and global makespan as performance indicators.

In the future, as a continuation of the reported work, a consensus-based approach will be investigated with the objective to allow some entities (products and/or machines) to converge towards a predefined state using the well-known consensus control theory. The predefined state will be computed such that the performances are improved in a distributed way.

Acknowledgements The authors gratefully acknowledge the financial support of the CPER 2007–2013 Structuration du Pôle de Compétitivité Fibres Grand'Est (Competitiveness Fibre Cluster), through local (Conseil Général des Vosges), regional (Région Lorraine), national (DRRT and FNADT) and European (FEDER) funds used to implement the TRACILOGIS platform.

Appendix: Simulation Run on the Basis of TRACILOGIS Test-Bed Platform

Negotiation production mode					Reactive production mode			
Run no.	Resource utilization rate in %		Lateness [M unit time]	Makespan [M unit time]	Resource utilization rate [%]		Lateness [M unit time]	Makespan [M unit time]
	M1	M2			M1	M2		
1	57.638	9.041	18.034	4.213	52.8558	8.1424	25.378	4.936
2	58.292	8.919	20.563	4.153	52.7938	8.1256	25.413	4.932
3	54.409	7.872	25.137	4.453	53.2695	7.9293	26.183	4.972
4	56.482	8.8647	14.991	4.214	52.7212	8.0879	25.17	4.919
5	54.979	8.3198	18.767	4.453	52.1597	8.3806	23.965	4.858
6	57.179	8.6526	15.604	4.126	53.1261	8.1643	25.148	4.956
7	57.499	8.7195	15.984	4.765	52.0837	8.2883	24.216	4.857
8	55.149	8.3472	17.913	4.543	53.3881	8.1176	26.32	4.985
9	55.840	8.6255	21.859	4.654	53.0012	7.9846	25.174	4.948
10	54.207	8.028	22.995	4.358	53.0811	7.917	26.063	4.955
11	57.298	9.1398	19.411	4.398	52.9325	8.1433	25.068	4.944
12	57.456	8.9456	20.467	4.239	52.2755	8.5515	23.91	4.868
13	57.792	8.9819	15.446	4.543	52.195	7.9588	23.867	4.859
14	55.994	8.6523	21.682	4.675	52.7988	7.991	25.49	4.933
15	57.559	9.0872	20.071	4.267	52.9869	8.0779	25.383	4.953
16	55.845	8.9843	21.084	4.567	53.022	8.2144	25.165	4.947
17	57.466	9.1792	15.188	4.132	52.9281	7.8771	25.335	4.944
18	61.161	9.5158	16.537	4.657	52.3901	8.0342	24.304	4.879
19	60.747	9.0764	17.322	4.155	53.1266	7.8721	25.679	4.963
20	57.256	8.6505	15.533	4.458	52.6351	8.0341	24.901	4.912
21	56.655	9.0554	16.523	4.358	52.1935	8.4481	24.784	4.872
22	59.807	9.6894	17.085	4.307	53.3609	8.1554	26.494	4.987
23	56.425	8.557	19.225	4.752	54.1766	8.233	23.611	4.821
24	60.195	9.367	15.771	4.218	52.3789	8.5047	24.273	4.885
25	61.952	10.00	12.808	4.182	52.8363	8.107	25.109	4.931
26	60.385	9.142	12.654	4.219	53.4946	7.87	26.454	5.007
27	57.468	8.928	21.127	4.517	53.0372	7.8231	26.012	4.953
28	56.720	8.679	18.539	4.502	52.9587	8.0193	25.127	4.943
29	63.178	10.12	10.019	3.987	52.2012	8.3483	24.096	4.870

(continued)

(continued)

Negotiation production mode					Reactive production mode			
Run no.	Resource utilization rate in %		Lateness [M unit time]	Makespan [M unit time]	Resource utilization rate [%]		Lateness [M unit time]	Makespan [M unit time]
	M1	M2			M1	M2		
30	57.286	8.441	15.721	4.558	52.3998	8.08	24.379	4.889
31	56.055	8.149	18.875	4.762	53.0937	8.0205	25.896	4.959
32	60.848	9.435	15.101	4.204	53.6773	7.9681	26.478	5.022
33	56.047	9.095	21.409	4.571	52.9313	8.3785	25.48	4.944
34	58.812	9.111	16.331	4.368	52.9335	7.8164	25.638	4.941
35	55.530	8.400	21.212	4.559	53.3417	7.9585	26.564	4.984
36	56.186	8.387	23.44	4.69	52.991	7.9121	24.917	4.945
37	61.971	9.591	14.073	4.183	52.4133	8.1897	24.136	4.886
38	58.199	8.641	17.288	4.423	53.6607	7.8973	26.543	5.018
39	57.131	9.104	19.597	4.473	53.1413	7.6211	25.444	4.968
40	56.563	8.648	23.027	4.69	53.311	8.5276	25.603	4.979
41	58.364	8.944	18.927	4.398	53.0403	8.338	25.335	4.951
42	57.425	8.831	15.744	4.523	52.9237	8.0495	25.628	4.944
43	57.482	8.072	22.401	4.746	52.9941	7.888	25.488	4.943
44	64.297	9.900	11.281	4.073	53.1654	8.1327	25.78	4.965
45	54.768	8.299	22.378	4.74	53.1247	7.8174	25.989	4.964
46	57.778	8.828	20.311	4.482	52.6639	7.9374	25.199	4.914
47	62.478	9.6444	14.099	4.134	52.8076	8.0659	25.518	4.934
48	59.474	9.4307	18.039	4.47	52.8131	7.9705	25.3	4.928
49	55.102	8.466	18.669	4.717	52.0553	7.9994	24.664	4.853
50	60.780	9.306	14.527	4.331	52.2629	8.3151	24.559	4.873
Total	2895.6	445.8	900.789	221.16	2644.1543	404.2846	1262.63	246.593
Average	57.912	8.917	18.01578	4.4232	52.883086	8.085692	25.2526	4.93186

References

1. Cardin, O., Trentesaux, D., Thomas, A., Castagna, P., Berger, T., El-Haouzi, H.B.: Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges. J. Intell. Manuf. 1–15 (2015)
2. Farid, A.: A review of holonic manufacturing systems literature. Technical Report, University of Cambridge, Institute for Manufacturing, UK (2004)
3. Wooldridge, M.: An Introduction to Multi-agent Systems. Wiley (2009)
4. Botti, V., Giret, A.: ANEMONA: A Multi-agent Methodology for Holonic Manufacturing Systems. Springer Science and Business Media (2008)
5. Vlassis, N.: A Concise Introduction to Multi-agent Systems and Distributed AI. MIT Press (2003)
6. Weiss, G.: Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press (1999)
7. Dimopoulos, Y., Moraitis, P.: Multi-agent coordination and cooperation through classical planning. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'06, pp. 398–402 (2006)

8. Zambrano, G., Pach, C., Aissani, N., Berger, T., Trentesaux, D.: An approach for temporal myopia reduction in heterarchical control architectures. In: 2011 IEEE International Symposium on Industrial Electronics (ISIE), pp. 1767–1772 (2011)
9. Tonino, H., Bos, A., de Weerdt, M., Witteveen, C.: Plan coordination by revision in collective agent based systems. Artif. Intell. **142**(2), 121–145 (2002)
10. Nagarajan, M., Sošić, G.: Game-theoretic analysis of cooperation among supply chain agents: review and extensions. Eur. J. Oper. Res. **187**(3), 719–745 (2008)
11. Aydogan, R., Baarslag, T., Hindriks, K.V., Jonker, C.M.: Heuristic-Based Approaches for CP-Nets in Negotiation. Springer (2013)
12. Rahwan, I., Ramchurn, S.D., Jennings, N.R., Mcburney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. Knowl. Eng. Rev. **18**(4), 343–375 (2003)
13. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proc. IEEE **95**(1), 215–233 (2007)

Part VII
Applications and Demonstrators

Smartness Versus Embeddability: A Tradeoff for the Deployment of Smart AGVs in Industry

Guillaume Demesure, Damien Trentesaux, Michael Defoort, Abdelghani Bekrar, Hind Bril, Mohamed Djemaï and André Thomas

Abstract In order to deploy AGVs in industry, it is mandatory to consider the tradeoff between smartness and embeddability. This paper aims at making the manufacturing research community more sensitive about this tradeoff and its consequences. Nowadays, AGVs are widely chosen by manufacturers to implement flexible material-handling systems which are necessary to cover the industrial requirements. However, many issues, presented in this paper, must be tackled to deploy these AGVs. A tradeoff-oriented procedure is proposed by considering these issues in flexible manufacturing system applications. Then, an approach is proposed to illustrate this procedure by providing simulation and experimental results. This approach is also used to roughly describe the smartness/embeddability tradeoff.

Keywords Automated guided vehicles · Flexible manufacturing systems Smartness · Embeddability

G. Demesure (✉) · D. Trentesaux · M. Defoort · A. Bekrar · M. Djemaï
LAMIH, CNRS UMR 8201, UVHC, 59313 Valenciennes, France
e-mail: guillaume.demesure@univ-lorraine.fr

D. Trentesaux
e-mail: damien.trentesaux@univ-valenciennes.fr

M. Defoort
e-mail: michael.defoort@univ-valenciennes.fr

A. Bekrar
e-mail: abdelghani.bekrar@univ-valenciennes.fr

M. Djemaï
e-mail: mohamed.djemai@univ-valenciennes.fr

G. Demesure · H. Bril · A. Thomas
Université de Lorraine, CRAN CNRS UMR 7039, Campus Sciences, 54506 Nancy, France
e-mail: hind.el-haouzi@univ-lorraine.fr

A. Thomas
e-mail: andre.thomas@univ-lorraine.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_30

1 Introduction

Day after day, the industrial requirements evolve and become stricter [17], where reactivity in the short term and adaptability to the market in the long term become harder to obtain. This can lead the industrials to reconsider their manufacturing plants by applying new approaches to cover these requirements. In this context, flexible manufacturing systems have several interests to consider automated guided vehicles (AGVs). Indeed, since the technological evolution in mechatronics, computer science and Information and Communication Technologies (ICT) allows improving the use of AGV while limiting their cost [19], the scope of reactive behaviour in the dynamic routing of AGVs is enlarged. In this paper, AGV-based flexible manufacturing systems are considered where products must be completed according to some manufacturing specifications. The AGVs, used as product transportation systems, are assumed to navigate freely on the production floor.

The use of AGVs in manufacturing plants allows improving their flexibility in terms of material-handling or routing, aiming at greater responsiveness to industrial requirements. However, it needs considering several AGV aspects for an efficient functioning of their plants. For example, the embeddability and the feasibility of design approaches must be tackled before the AGV deployment in industry. The feasibility allows proving that an approach is technically feasible and economically profitable. The embeddability [11] refers to the capacity to embed enough communication, computational and energy devices in AGVs. Based on the holonic paradigm, an AGV can be considered as a resource holon in a HMS since decisional capabilities are embedded [3]. The embeddability depends on the smartness level of their functions. Indeed, having a high-level of smartness means that the embedded functions have the capacity to deal with complex situations. However, the AGVs may not have the capacity to apply them correctly in short time, leading to computational overload which impacts the feasibility. Conversely, because of limited available processing capabilities, the AGVs may only be able to deploy simple functions where their lack of smartness prevents them from dealing with complex situations and being used at their full potential. Thus, one can see the AGV tradeoff between their smartness and their embeddability.

To deal with this tradeoff, the manufacturing research community and the industrial one design their approach differently. The industrial community deals more with the embeddability by designing simple embeddable functions with little attention being paid of the AGV smartness. Conversely, the manufacturing research community tends to design complex approaches by improving this smartness. However, the complexity to solve their smart approach is hardly compensated by the technical evolution. For example, the navigation is one of AGV functions which could be hard to embed. Indeed, the navigation tools are often complex since they need to be smart enough to prevent unexpected situations (e.g. deadlock [8], conflicts [7], local minima [3] …). Therefore, this paper is focused on approaches which include a navigation function, especially due to its major impact on the tradeoff.

Among the several navigation tools, motion planners are widely used [13], allowing the generation of collision-free trajectories between two configurations. Since the AGV navigation is related to the production performances, mathematical programming seems to be an interesting tool [5]. However, the motion planning problems are NP-Hard [4], leading to the need reduce the computational time with meta-heuristic algorithms [14] or with discretization of the navigation area [16]. Even if the navigation may prevent the AGV deployment in industry, some approaches have been proposed in indoor environments. The Kiva system [20], which is currently used in Amazon warehouses [9], uses a standard implementation of A^* algorithm to plan paths. In [12], the AGV navigation is solved using a D^* algorithm coupled with other functions.

The main purpose of this paper is to make the designer of AGV-based flexible manufacturing systems more sensitive about this existing tradeoff. Based on a design for "x" paradigm [10], designing for tradeoff is more suitable than only considering the smartness or the embeddability. Hence, the designers of AGV-based flexible manufacturing systems need to consider new approaches or reconsider their previous ones to deal with this tradeoff. Therefore, this article can be viewed as a basis where the main issues are presented and are related to a procedure representing the several design and test steps. An illustrative example is provided by proposing an approach tested in simulation and experimentally environment. This example highlights the consequences of the smartness/embeddability tradeoff.

The paper is organized as follows. Section 2 presents the main issues that designers may tackle using AGV functions. A design procedure, allowing to highlight the tradeoff and consequences, is proposed in Sect. 3. The illustrative example with results and discussions are provided in Sect. 4.

2 Issues Which May Impact the Tradeoff

In order to deal with the mentioned tradeoff, it is necessary to introduce the main issues that AGV may solve using smart functions. Firstly, collision-free trajectories between resources are required since AGVs navigate freely on the production floor. These trajectories (paths) have to be feasible due to the physical behaviours and constraints. The disturbances and discrepancies must be taken into account to prevent deviation from the planned trajectory. Furthermore, all issues related to localization, mapping and actuators/sensors must be tackled. Also, communication devices are required to send information in short time between AGVs and the higher level of control (e.g. resources, supervisor…). The energy is another issue to tackle since the battery has limited capacities. Indeed, the battery charge/discharge may change the decisions that AGVs make according to the task they have to perform.

Secondly, several other issues are related to the AGV environment since they navigate in a flexible manufacturing system. For example, manufacturing disturbances such as machine breakdown may occur at any time which impacts the AGV decisions. As each AGV computes its own functions, the control architecture is fully or partially distributed. In spite of improving the reactivity, this architecture is more complex in structure and organization and may lead to myopic behaviours [17]. To remove these drawbacks, the control architecture must be adapted (by including a supervisor for example) and the interactions between the different entities (defined in [17]) must be tackled to prevent decision conflicts [2].

The last issues are related to the industrial environment and the AGV deployment. Human factors must be taken into account because human workers may impact the AGV-based flexible manufacturing system. For example, humans may work on some resources or AGVs for maintenance purpose. The humans must be able to understand the AGV behaviours as well as to take control of the system in case of failure. Therefore, safety functions, allowing to prevent collisions with humans, and AGV/human cooperation mechanisms must be embedded in the AGVs.

All the issues mentioned above are related to the smartness of AGVs. However, the embeddability issue may prevent them to improve their smartness. Hence, the number of functions that they cover may be limited, due to technically and/or economically reasons. It may lead to reconsider their functions (e.g. the number of functions and their role) in order to reach a good tradeoff. This tradeoff can be seen as a balance where the smartness and the embeddability, respectively depending on the AGV functions and their device capabilities, are on each side of the balance as shown in Fig. 1.

One can see on this figure that improving the smartness increases the weight on the left side of the balance. To compensate, the embeddability has to be improved by adding some weights in the right side. However, it impacts the deployment cost since AGV with better capabilities are more expensive. Thus, this cost becomes an important issue since their deployment needs to be economically profitable, allowing a good return on the investment. From these statements, the embeddability is limited by this cost, preventing the improvement on the smartness of AGV functions.

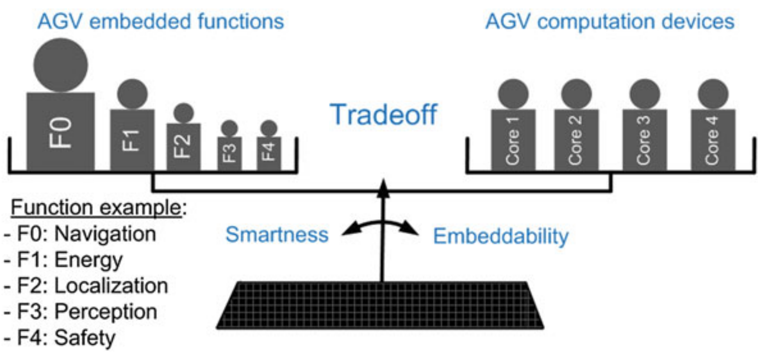


Fig. 1 Tradeoff balance between smartness and embeddability

3 Proposed Tradeoff-Oriented Procedure

The design of AGV approaches in flexible manufacturing system depends on production objectives and specifications. To cover these requirements, the AGVs must tackle one or several issues by using embedded functions. To keep the AGV benefits, it is often recommended to follow a design procedure while thinking about the next steps where some issues may occur. In Fig. 2, a tradeoff-oriented procedure is proposed by providing several design steps and tests until the AGV deployment.

Each step has its own role but may depend on other steps. The role of each step is described as follows:

- *AGV objective and assumption setup*: In this step, the problem is set for each AGV by defining their objective. Furthermore, several assumptions can be proposed to make the problem simpler. During the next steps, these assumptions may be reconsidered to take more issues into account.
- *AGV function assignment and architecture design*: This step is used to define the different functions of the AGV. The functions that AGV do not cover must be tackled by other entities (e.g. supervisor). Thus, the control architecture, where the interactions between the different entities are studied, must be designed to prevent conflicts between them.
- *Detailed design of AGV smart functions*: The AGV functions, including the navigation one, are detailed in this step. If the tradeoff between smartness and embeddability is not reached, some functions may be exchanged with the manufacturing high-level of control.

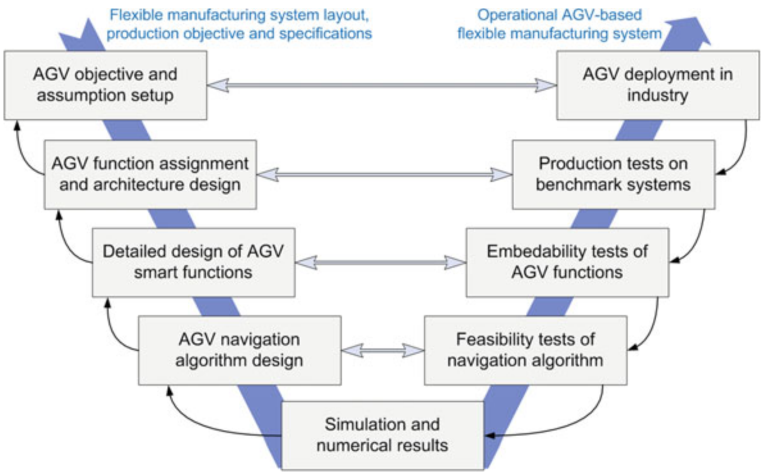


Fig. 2 Main steps of AGV design procedure until the deployment in industry

- *AGV navigation algorithm design*: After defining the functions, the navigation tools of AGVs must be designed such as the motion planner, the control strategy, the used algorithm (e.g. A^*, meta-heuristics, …)
- *Simulation and numerical results*: These first results allow showing a preview about the feasibility, the smartness and the performances of the designed approach. When the results lead to infeasibility, an in-depth analysis must be done. This analysis allows finding the step(s) at which modifications are required. If the AGV smartness or the performances are not good enough, the functions both from the AGV and from other entities must be modified or improved.
- *Feasibility tests of navigation algorithms*: In this step, the navigation algorithms are experimentally tested to check if the trajectories, provided by the navigation tools, are feasible. If not, the AGV navigation algorithms must be improved by reconsidering the two previous steps.
- *Embeddability tests of AGV functions*: After coding and embedding all the AGV functions, the embeddability is checked. An experimental scenario must be designed to test the overall functions by letting the AGVs do their mission. If they do not have the capacity to do it, the smart functions of AGVs must be redesigned.
- *Production tests on benchmark systems*: When embeddability is checked, the AGV functions can be tested on higher instances. A full production scenario (e.g. benchmark [18]) is designed. However, some functions could be missing to cover other issues such as global optimization or energy. Thus, these functions need to be included in the AGVs or in the manufacturing high-level of control and previous steps must be tackled again.
- *AGV deployment in industry*: To carry out this step, the different issues related to the industrial environment must be tackled. It can lead to reconsider some assumptions, made to simplify the designed approach. Other systems, such as the supervisor, have to be designed to take human factor into account. For example, user-guided interface can be used, allowing human workers to take control of the manufacturing plant or to understand the AGV behaviour. Moreover, the cost must be evaluated, leading to reconsider previous steps to reduce the AGV deployment cost.

In order to reach a good tradeoff, the idea is not to follow the procedure step by step but rather to follow a design for tradeoff procedure, thinking about the AGV capability and the cost required to obtain these capabilities. Indeed, using the balance paradigm (see Fig. 1), some weights are positioned on each side of the balance. On one side, the weights correspond to the AGV functions, depending on their complexity. On the other side, the weights are related to the AGV computational capabilities where for example, one weight may correspond to one processor core. Heavier weight gives higher AGV capabilities but their cost are also higher. Therefore, to take the design for tradeoff aspect into account, it should be suitable to deal with the AGV capabilities and their consequence it terms of cost before starting designing their smart functions. After defining these capabilities, the AGV functions could be designed until the tradeoff is respected. It can lead to reduce the function smartness or to remove some functions in AGVs. Hence, cooperation mechanisms [6] with other entities of the flexible manufacturing system become useful to outweigh the AGV smartness according to their capabilities.

4 Illustrative Example

To illustrate the proposed tradeoff-oriented procedure, an example is proposed in this section where AGVs navigate on a manufacturing production floor. This illustrative example is tested by giving some simulation and experimental results. The steps of the procedure, in Fig. 2, are roughly described and the tradeoff is discussed according to the provided AGV capabilities.

AGV design steps: In the proposed approach, the production objective is to execute several products, transported by AGVs. Each AGV transports only one product from a resource to another one until the product is completed. For each product, transported by one AGV, an operational sequence must be followed where operation are tackled one by one. For each operation, some production specifications are provided to the AGV by the manufacturing operation and resource management (M.O.R.M.) level. These specifications include a due date for which the operation must be completed and a set of resources at which the operation may be performed. Moreover, the resources have queuing capacities, assumed infinite at first, and AGVs in queue must wait to perform their operation when the resource is unavailable (e.g. performing an operation of other AGV). The resource disturbances and the energy issues are not tackled to make the problem simpler.

Since the flexible manufacturing system layout, the production objective and the different specifications are given, the AGV design steps (i.e. left side of the procedure provided in Fig. 2) can be described. The AGV objective is both to compute the best resource and to generate a collision-free trajectory towards this resource. The best resource is chosen by minimizing the time to complete the ongoing operation and depends on the transportation time, the waiting time and the processing time of the resource. Minimizing the completion time is equivalent to complete the operation at the soonest date. In terms of assumptions, all AGVs have the same physical behaviours where their velocities are bounded and they are supposed to know their position at any time as well as the position of the resource they are moving towards. They have a limited communication range and are called neighbours when they are able to communicate. Moreover, they are able to communicate with the higher level of control, to receive help or specifications and transmit feedbacks about their configuration.

The different functions assigned to the AGVs are mainly focused on their navigation. The motion planning function, combined with the scheduling one, allows selecting a resource at which the operation will be performed while generating a collision-free trajectory towards this resource. The tracking strategy function is in charge of following the planned trajectory in spite of inherent discrepancies and disturbances. The control architecture, where all AGV functions are highlighted, is given in Fig. 3: the AGV may interact with a supervisor which allows helping

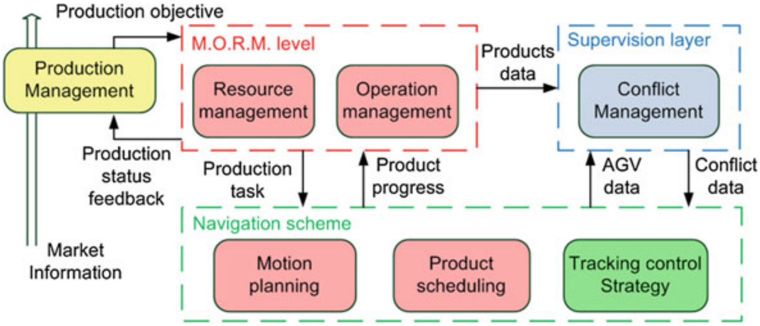


Fig. 3 Control and decision architecture of the illustrative example

their navigation by solving the conflicts using performance-based priority negotiation. Moreover, it allows preventing their myopic behaviours since the AGVs only communicate with their neighbours. The supervisor has the role to solve the different conflicts. At first, it defines the arrival order of AGVs if they are moving towards the same resource. Furthermore, the AGV applies the scheduling function only if the supervisor allows it. The last supervisor function allows AGV to anticipate the collision by providing them some variable areas where big conflict may occur. It allows AGV to avoid these areas before reaching them, reducing the complexity of the AGV motion planner since big conflict areas are prevented.

For the detailed design of AGV functions, the motion planner uses physical constraint (e.g. velocity bound) and temporal constraint (e.g. due date) to compute a collision-free trajectory towards a selected resource. This planner is applied gradually over time and is divided into two steps. The first step, where the resource is chosen, is used as a global planner where AGVs must decide their intention by planning a presumed trajectory to avoid the conflicts given by the supervisor. The second step is local since each AGV uses the presumed trajectories, both its own and its neighbours' ones, to compute its final collision-free trajectory. One can notice that each AGV only avoids its neighbours having higher priority. It means that the AGV with the highest priority does not need to avoid others.

In terms of algorithm design, mathematical programming is used for each part (global and local) of the motion planner since it is related to the manufacturing performances. For embeddability purpose, the solving algorithms have to provide the trajectories in short time since they are computed over time. To reduce the computational costs, a Particle Swarm Optimization (PSO) is used due to its relatively fast convergence and global search character [15].

AGV test steps: Since the design steps have been given, the approach has to be tested to follow the procedure given in Fig. 2. To provide the simulation and numerical results, the AGV physical constraints are required. The AGVs are here represented by Lego Mindstorms robots where their maximum allowed velocity is 0.15 m/s. Their communication range and safety distance are respectively set to 0.4 and 0.2 m.

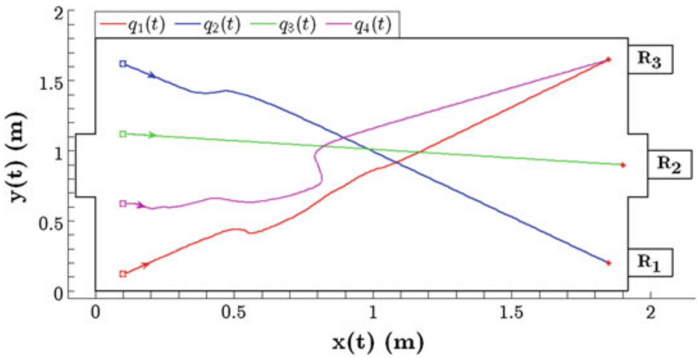


Fig. 4 Simulation of the illustrative example: AGV planned trajectories

Table 1 Agent initial parameters of the proposed scenario

	Initial position	Operation due date (s)	Processing time (s)	Resource possibilities
AGV 1	[0.1, 0.125]	76.1	5.4	$\{R_3\}$
AGV 2	[0.1, 0.625]	30.8	4.3	$\{R_1\}$
AGV 3	[0.1, 1.125]	23.8	5	$\{R_2\}$
AGV 4	[0.1, 1.625]	60.2	4.7	$\{R_2, R_3\}$

The number of robots (called *AGV i*) is limited to 4 (i.e. $i = 1, 2, 3, 4$) and they navigate in a reduced area where three virtual resources (called R_c, $c = 1, 2, 3$) are positioned (as shown in Fig. 4). The positions (x_c, y_c) of resources R_c are set to (1.85, 0.2), (1.9, 0.9) and (1.85, 0.65), respectively. For this scenario, the several AGV initial parameters can be found in Table 1. When the scenario starts, AGV 3 has the highest priority while AGV 4 has the lowest one.

For the simulation results of the proposed scenario, the AGV planned trajectories are provided in Fig. 4. One can see that AGV 3 does not avoid any other AGVs due to its highest priority. Conversely, AGVs 1 and 4 must adapt their trajectories to avoid others and AGV 2 needs to avoid AGV 3.

The approach is feasible in simulation since the velocity and collision avoidance constraints are fulfilled according to Fig. 5. Moreover, Table 2 shows that the operations of products transported by AGVs are completed before the provided due dates. During the navigation, one can see that AGV 4 changes its chosen resource. It implies that AGV 1 has lower priority than AGV 4. Therefore, the AGV smartness is proved since they are able to apply their function in spite of their specifications they have to fulfill. Indeed, collision-free trajectories are planned by the AGVs and the scheduling function is applied (AGV 4) to improve its product performances. Moreover, the supervisor allows them to prevent unsolvable conflict arisen from myopic behaviours.

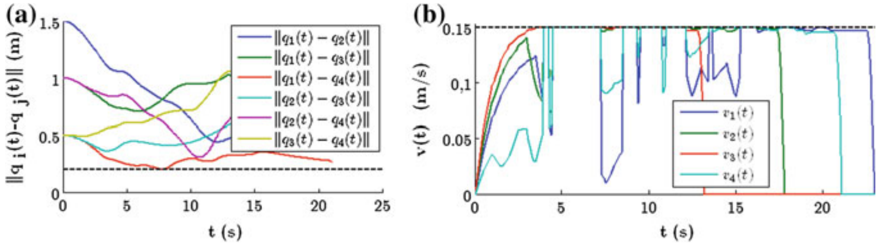


Fig. 5 Simulation of the illustrative example: AGV distances (**a**) and velocities (**b**)

Table 2 Numerical results of the proposed scenario

	AGV 1	AGV 2	AGV 3	AGV 4
Final time (s)	25.44	17.76	13.14	21.02
Completion time (s)	31.12	22.07	18.14	25.72
Chosen resource	R_3	R_1	R_2	R_3

The next step of the procedure is focused on the feasibility tests of navigation functions. Thus, the trajectories are tested by the Lego Mindstorms robots for this scenario. The feasibility of the trajectories is demonstrated experimentally as shown in the video[1] where all explanations are provided.

To test the embeddability of AGVs, it is mandatory to consider the provided robots' capabilities. These robots do not have the same capabilities as industrial AGVs, which are able to cover different functions [12]. Thus, all of designed functions cannot be embedded in AGVs to prevent bad tradeoffs as shown in Fig. 6a and b. To reach a good tradeoff, the AGV functions must be reduced to the tracking one (see Fig. 6c). It means that the combined motion planning/scheduling functions are computed by another entity, such as the supervisor, and then transmitting to the AGVs.

Using the Lego Mindstorms robots, the production tests are hard to obtain since they do not have capabilities to plan their own trajectory. Indeed, planning a trajectory gradually over time for each AGV is computationally expensive, preventing dealing with manufacturing uncertainties such as machine breakdown. To perform these production tests, reconsidering the used AGVs seems to be more appropriate. Therefore, the AGV capabilities have to be improved to use their functions at full potential while maintaining a good tradeoff. Thus, the weights of the tradeoff balance changes, as shown in Fig. 6d, by proposing some required AGV capabilities.

In order to deal with our proposed approach, the AGVs must be able to compute their own functions simultaneously. Thus, improving their capabilities is required by using multi-core processors for example. Using these processors, the computational time could be reduced and other tools, such as parallel meta-heuristic [1], may be proposed to solve motion planning problem. Indeed, since a resource must be

[1]Available at: https://www.youtube.com/watch?v=KUY7cBRx9vIŹfeature=youtu.be.

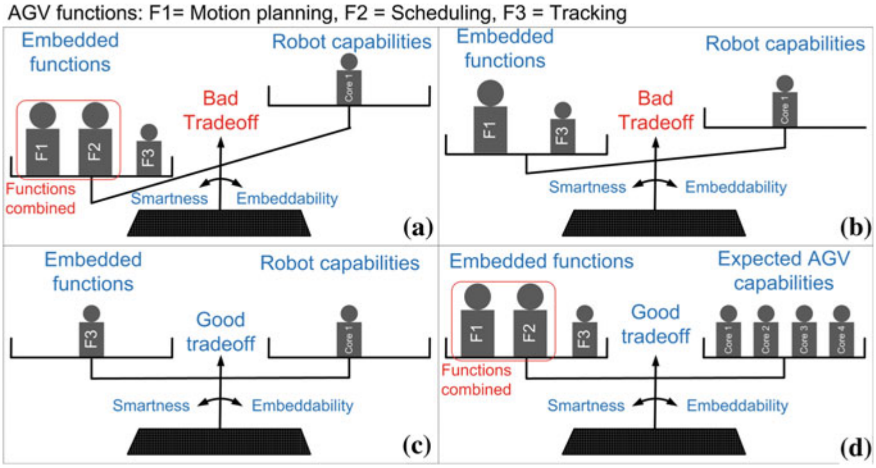


Fig. 6 Tradeoff balance for the proposed approach: **a** all AGV functions, **b** scheduling function removed, **c** tracking function only, **d** all functions with better capabilities

selected, a trajectory for each resource could be computed in parallel. Moreover, the AGV capabilities must consider other simple functions such as the safety or energy ones which are not tackled in the proposed approach. Therefore, using multi-core processors is more suitable where each core can be represented as a weight in the embeddability side of the balance (see Fig. 1). Nowadays, quad-core processors are easily embedded (e.g. in smart phones) and become less expensive than in the last decades. Thus, their use as AGV device may be economically profitable.

5 Conclusion

In this paper, a tradeoff between smartness and embeddability is discussed for the use of AGVs in manufacturing plants. Several issues, which may be solved using AGV functions and have a possible influence on this tradeoff, are presented. The objective allows making the designers of AGV-based flexible manufacturing systems more sensitive about this tradeoff and its consequences. A tradeoff-oriented procedure is advisable to balance the smartness according to the AGV functions and its capabilities. An illustrative example on an AGV-based flexible manufacturing system is proposed where simulation and feasibility tests are provided to highlight the importance of this tradeoff for the AGV deployment in industry. To quantify and ease the analysis of this tradeoff, depending on the company's culture and methods, the designer could use some methods from the quality domain, like Value Analysis Method, Kano Model or Quality Function Deployment (QFD). Consequently, this first work leads to many perspectives from: (i) methodological view: how to choose efficient AGVs depending on requirements and how to adapt them to deals with organization or requirements changes and (ii) domain applications view: the introduction of AGVs manufacturing processes opens new ways to organize the shop floor. It could be interesting, for example, to evaluate which physical element is associated to AGV: the product or the resource?

References

1. Alba, E.: Parallel Metaheuristics: A New Class of Algorithms, vol. 47. Wiley (2005)
2. Bader, M., Richtsfeld, A., Suchi, M., Todoran, G., Holl, W., Vincze, M.: Balancing centralised control with vehicle autonomy in AGV systems for industrial acceptance. In: 11th International Conference on Autonomic and Autonomous Systems (2015)
3. Berger, T., Sallez, Y., Trentesaux, D.: Open control of FMS and its application to potential field. In: 42nd Conference on Manufacturing Systems (2009)
4. Canny, J.: The Complexity of Robot Motion Planning. MIT Press (1988)
5. Defoort, M., Veluvolu, K.C.: A motion planning framework with connectivity management for multiple cooperative robots. J. Int. Robot. Syst. **75**(2), 343–357 (2014)
6. Demesure, G., Defoort, M., Bekrar, A., Trentesaux, D., Djemaï, M.: Cooperation mechanisms in multi-agent robotic systems and their use in distributed manufacturing control: issues and literature review. In: IECON 2014-40th Conference of the IEEE, pp. 2538–2543. IEEE (2014)
7. Demesure, G., Defoort, M., Bekrar, A., Trentesaux, D., Djemaï, M.: Navigation scheme with priority-based scheduling of agents: application to AGV-based FMS. J. Int. Robot. Syst. 1–18 (2015)
8. Erol, R., Sahin, C., Baykasoglu, A., Kaplanoglu, V.: A multi-agent based approach to dynamic scheduling of machines and AGVs in manufacturing systems. App. Soft Comput. **12**(6), 1720–1732 (2012)
9. Gilmour, K.: Amazon warehouse, Amazon adventure. Int. Mag. (2003)
10. Kuo, T.C., Huang, S.H., Zhang, H.C.: Design for manufacture and design for 'x': concepts, applications, and perspectives. Comput. Ind. Eng. **41**(3), 241–260 (2001)
11. Malinowski, A., Yu, H.: Comparison of embedded system design for industrial applications. IEEE Trans. Ind. Inf. **7**(2), 244–254 (2011)
12. Martínez-Barberá, H., Herrero-Pérez, D.: Autonomous navigation of an automated guided vehicle in industrial environments. Robot. Comp.-Int. Man. **26**(4), 296–311 (2010)
13. Masehian, E., Sedighizadeh, D.: Classic and heuristic approaches in robot motion planning—a chronological review. World Acad. Sci. Eng. Technol. **23**, 101–106 (2007)
14. Masehian, E., Sedighizadeh, D.: A multi-objective PSO-based algorithm for robot path planning. In: IEEE International Conference on Industrial Technology (ICIT), pp. 465–470. IEEE (2010)
15. Saska, M., Macaš, M., Přeučil, L., Lhotska, L.: Robot path planning using particle swarm optimization of ferguson splines. In: IEEE Conference on Emerging Technologies and Factory Automation, pp. 833–839. IEEE (2006)
16. Thyagarajan, R.: A motion control algorithm for steering an AGV in an outdoor environment. Ph.D. thesis, College of Engineering, Osmania University, India (2000)
17. Trentesaux, D.: Distributed control of production systems. Eng. Appl. Art. Int. **22**(7), 971–978 (2009)
18. Trentesaux, D., Pach, C., Bekrar, A., Sallez, Y., Berger, T., Bonte, T., Leitão, P., Barbosa, J.: Benchmarking flexible job-shop scheduling and control systems. Cont. Eng. Pract. **21**(9), 1204–1225 (2013)
19. Ueda, K.: Emergent synthesis approaches to biological manufacturing systems. In: Digital Enterprise Technology, pp. 25–34. Springer (2007)
20. Wurman, P.R., D'Andrea, R., Mountz, M.: Coordinating hundreds of autonomous vehicles in warehouses. AI Mag. **29**(1), 9 (2008)

H^2CM-Based Holonic Modelling of a Gas Pipeline

Carlos Indriago, Latéfa Ghomri and Olivier Cardin

Abstract A gas pipeline is a relatively simple physical system, but the optimality of the control is difficult to achieve. When switching from one kind of gas to another, a volume of useless mixture is generated. Therefore, the control needs to both respond to the demand and minimize the volume of lost gas. In case of stable and perfectly known demand, scheduling techniques can be used, but in other cases, calculation times are incompatible with an industrial application. This article introduces the application of H^2CM (Holonic Hybrid Control Model) generic architecture on this specific case. The study case is extensively presented. Then, the defined holonic architecture (H^2CM compatible) is detailed, and the role and functions of each holon are presented. Finally, a tentative general control algorithm is suggested, which gives an insight on the actual algorithms that will be developed in perspective of this work.

Keywords Hybrid dynamic systems • Holonic hybrid control model
Pipeline transport system

C. Indriago
Centro de Investigación de Procesos-CENIPRO, Universidad Politécnica Antonio José de Sucre, Barquisimeto, Venezuela
e-mail: cindriago@unexpo.edu.ve

L. Ghomri
Manufacturing Engineering Laboratory of Tlemcen, Department of Electrical Engineering and Electronics, Abou-Bekr Belkaïd University, Tlemcen, Algeria
e-mail: ghomri@mail.univ-tlemcen.dz

O. Cardin (✉)
Laboratoire des Sciences du Numérique de Nantes, LUNAM Université, IUT de Nantes Université de Nantes, LS2N UMR CNRS 6004 2 avenue du Prof. Jean Rouxel, Carquefou, France
e-mail: olivier.cardin@ls2n.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_31

1 Introduction

Hybrid dynamic systems (HDS) are dynamic systems integrating explicitly and simultaneously continuous systems and discrete event systems. They require for their description the use of continuous time models, discrete event models and the interface between them [1]. The hybrid character of the system either owes to the system itself or to the control applied to this system. Typical examples of such systems are communication protocols, manufacturing systems, transportation systems, power electronics, etc.

Modelling, analysis and control of HDS are crucial concerns. Two of the most important formalisms for the modelling and analysis of HDS are hybrid automata [2] and hybrid Petri nets [3]. Hybrid automata can consider any continuous dynamics in a location, and the commutation from one location to the other one is synchronized by a discrete event. It is then possible to model any type of system. The main modelling drawback of the hybrid automata is the explosion of the number of locations in case of real life systems. To overcome this problem, hybrid Petri nets consider a state as a marking with a continuous and a discrete part. They provide very compact and readable models very useful for engineers. However, in order to perform a formal analysis, it is necessary to come back to the hybrid automata which are known for their analysis power. This analysis requires the construction of the reachable state space. This operation is all the more complex because the discrete part is strongly nonlinear and the time is often non-deterministic. The calculation algorithms of the reachable state space only terminate under very restrictive constraints, for example for timed models or some linear hybrid automata where the continuous dynamics is constant. The reachable state space is described by a set of inequalities over the state variables, thus allowing both the performance analysis of the system and the synthesis of the control.

Most realistic formal approaches consider either continuous approaches with few commutations or discrete approaches with a very poor continuous dynamics (clocks). Since these systems are strongly nonlinear, any change in one or several parameters often forces us to completely redo the study of the problem. When the HDS becomes more complex, the analysis tools also become more complex turning them into loosely flexible systems with high calculation times, which do not react fast enough to unexpected events. In order to provide flexibility to the HDS, the researchers have studied the possibility of implementing flexible control architectures on complex dynamic systems [4, 5].

Holonic Hybrid Control Model (H^2CM) [5] is a holonic architecture developed with the aim of giving flexibility to HDS control and is based on the holonic architecture of discrete systems called PROSA [6]. It is composed of three basic holons:

- The product holon, which has all the information of the product;
- The resource holon, which is an abstraction of the resource;
- The order holon, which takes the information of both holons and generates the scheduling of the services to implement.

An example of a complex HDS is the pipeline system [7], which is a very important transport system that guarantees a regular supply of products and a rapid adaptation to market demand, thus significantly reducing costs and delays of product transportation. Pipeline systems are continuous operation systems that work with several products, resulting in a contaminated mixing zone on the contact of two products that are transported sequentially. Therefore, a greater number of batches transported will produce a larger number of contaminated product batches. With the objective of minimizing contaminated product batches, optimization methods are used that generate the batch sequence scheduling to be transported, with the restriction that these methods have high calculation times and are not flexible to changes in transport demand.

The objective of this article is to propose the implementation of the H^{2}CM architecture to a pipeline transport system in order to provide operational flexibility in the phase of changes in the demand for products, keeping the optimization criteria of the generation of contaminated product. This preliminary study extends the performance evaluation of H^{2}CM that was made on a water tanks system [8] with the notion of switching costs (contaminated product volume) and the dynamics of the pipeline (delay between the switch and the final tank filling).

The study case is extensively presented in the next section. Then, the defined holonic architecture (H^{2}CM compatible) is detailed, and the role and function of each holon are presented. Finally, a tentative general control algorithm is suggested, which gives an insight on the actual algorithms that will be developed in perspective of this work.

2 Case Study: Pipeline Presentation

2.1 Description of the ASR Multi-product Pipeline

The transportation of fuels by pipeline is increasingly spread throughout the world. This is explained by an increase in the quantities of transported products. This situation requires companies to further develop their logistics. It is in this objective that the Algerian oil companies have an investment program aimed at securing the country's petroleum products through an intelligent network of pipelines, responding to the real need of the different zones of the country.

The transportation by pipeline contributes both to the reduction of costs, delivery times, road traffic and also ensures mass transport respecting the environment with the most security. If pipelines did not exist, it would be inevitable to have thousands of trucks and railcars that circulate on roads, highways and railways to carry out the same transport.

The current technology is oriented towards multi-product pipelines. The latter has the disadvantage of creating a mixing or a contaminated product zone between two products in contact (Fig. 1), which circulate sequentially in the pipeline.

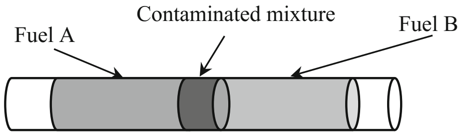


Fig. 1 Typical sequence with two products and one contaminated area

The contaminated product is generated at each contact of two different products of fuels. So a sequence of several batches involves proportionately several batches of contaminated product, requiring a large space for their storage.

In this paper, we focus our study on a typical pipeline of the National Petroleum Algerian Company (SONATRACH); it is the multi-product ASR pipeline (Abbreviation for the three cities: Arzew, Sidi Bel Abbès and Remchi in the west of Algeria, through which the pipeline passes). The pipeline transports fuel from the Arzew refinery to the storage and distribution sites of Sidi Bel Abbes and Remchi. They feed the regions West and South-West of Algeria in fuels (diesel fuel and gasoline). The demand of the region leads to the introduction into the ASR pipeline of important sequences of several batches to meet the needs of the region. This leads to numerous interfaces, zones of birth of the mixtures and high levels of the contaminated product stock at the depot of Remchi.

In view of the contamination constraints and the high demand of fuel, it will be interesting to study the optimization of the multi-product fuel transport. The objective will therefore be the minimization of the contaminated product stocks on one hand, and the satisfaction of the demands of the two distribution depots of pure products on the other hand.

2.2 Physical Data of the Pipeline

The multi-product ASR pipeline is located in the west of Algeria. Its profile extends over a length of about 168 km, from the refinery of Arzew passing by Sidi Bel Abbes depot and arriving at the final depot of Remchi.

The pipeline receives the liquid fuels from the Arzew refinery and supplies the storage and distribution depots in Sidi Bel Abbes and Remchi (Fig. 2).

Table 1 shows the storage capacities of each depot in the different fuel categories.

In Table 2 we represent the daily demand at the two depots level.

Figure 3 shows the overall structure of the Arzew, Sidi Bel Abbès and Remchi (ASR) pipeline, including the Arzew refinery and the two distribution depots in Sidi Bel Abbès and Remchi.

Fig. 2 Longitudinal profile of the multi-product pipeline ASR

Table 1 Capacities, initial volumes and safety stocks of products and contaminated products in the two depots

Fuels	Sidi Bel Abbes depot			Remchi depot		
	Capacity [m^3]	Initial stock [m^3]	Security stock [m^3]	Capacity [m^3]	Initial stock [m^3]	Security stock [m^3]
Diesel	6000	814	1200	22000	5572.4	4400
Pure gasoline	1700	809	340	9500	3394.6	1900
Super gasoline	450	196	90	1000	996	200
Unleaded gasoline				5000	3284.7	1000
Contaminated product type 1				500	396.5	
Contaminated product type 2				500	405	

Table 2 Daily demand in the two depots

Product	Daily demand	
	Sidi Bel Abbes depot	Remchi depot
Diesel	1200	3000
Pure gasoline	80	150
Super gasoline	400	800
unleaded gasoline		150

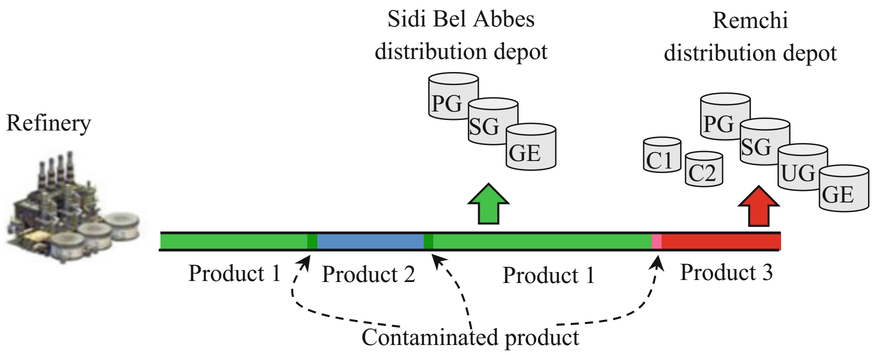


Fig. 3 Structure of the ASR system

3 Holonic Modelling

3.1 H^2CM Overview

The H^2CM generic architecture is based on the three basic holons of PROSA, introduced in Fig. 4a. Two main features can be highlighted:

1. Each resource is granted with an order and a product along its life. The order holon is in charge of the resource monitoring whereas the product holon is in charge of the recipe to be applied on the actual product. The content and objectives of the order and product holons are constantly evolving, but the structure remains constantly the same.

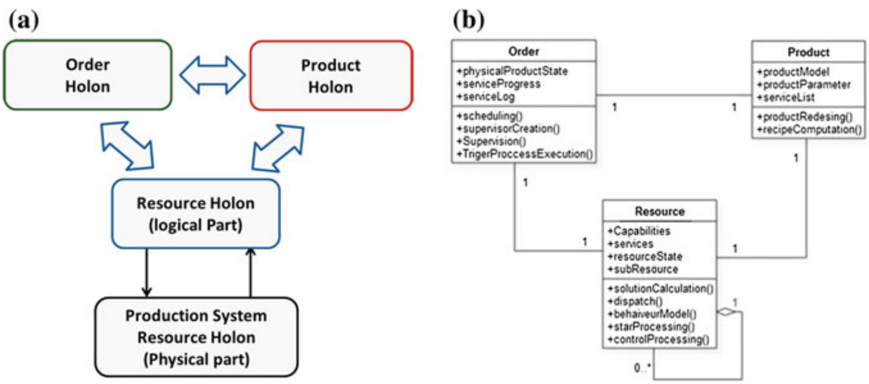


Fig. 4 **a** H^2CM basic holons **b** H^2CM diagram

2. A recursivity link is present on the resource holon, see Fig. 4b. Indeed, each compound resource can be fractally decomposed into one or several holarchies, comprising one or several resources and their associated order and product holons. The aggregation relations created here can be changed along the working of the system; holarchies can be created and destroyed online.

The order holon is globally responsible for the scheduling of the system. It is very representative of the dual working of every holons: one part is dedicated to the negotiation mechanisms in look-ahead mode in order to determine the future scheduling and holarchies of the system, while the other part of the holon is dedicated to the system in real time mode through the application of the scheduling resulting from the previous negotiations.

The product holon is dedicated to store and communicate the products recipes which are deduced from a Master Recipe. In H^2CM, this Master Recipe can be defined as a generic recipe, i.e. a sequential list of operations to be applied to the product to obtain a final product from raw materials (BOM—Bill of Materials), from which the actual recipe can be derived according to the conditions of the system. In HDS context, a service-oriented specification, as proposed in [9, 10], is well suited for the product specification. The distinction made in this article, with respect to the definition used in [9, 10], is that the parameters and variables of the service can be continuous or discrete.

By nature, HDS are large systems, constituted of many components. Therefore, a lot of resource holons are necessary to control the system. Figure 4b defines a recursivity of the resources, so the smallest resource holon to be defined is called atomic resources and can be expressed as “the maximal aggregation of elements whose system of differential equations can be inversed in a short delay relatively to the dynamics of the system”. Considering compound resources, the negotiation mechanism is meant to determine the best solution recursively, interrogating the aggregated resources until atomic ones. Resources holons are part of the negotiation mechanisms, and their function is to evaluate and transmit to the order holon the best possible variables values to obtain the desired function and services. Resource holons also have the responsibility for the devices online control, i.e. the role of controllers of the system.

Another specificity of the resource holon is its structure. Classically, it is composed by a physical part and a logical part, see Fig. 4a. The physical part is represented by the shop floor. The logical part of the resource holon is an abstraction of the physical part and contains the conversion models from continuous states to discrete states and vice versa. The models used are hybrid models that change their state using threshold levels of continuous variables. The abstraction of physical part can be developed by any industrial computer with communication capability.

3.2 Description of Holons and Services in the Case of a Pipeline

The work developed in this paper focuses on the multi-product ASR pipeline linking the Arzew refinery to the storage and distribution centers of Sidi Bel Abbes and Remchi. The latter deserves the centres in fuels namely: Diesel, Super Gasoline, Unleaded Gasoline and Normal Gasoline. From the holonic point of view, the ASR pipeline will be divided into three composite holons (see Fig. 5) which will offer three different types of services.

The first composite holon to be defined is the Arzew refinery, composed by the finished product tanks system and the pipe and pump system. The service offered by the holon of the Arzew refinery is the supply of fuel products. Looking at Fig. 6, the refinery resource holon has an associated product holon and an order holon. The product holon holds contaminated product information, such as the volume generated in each product mix and the variation of product density during mixing. The order holon has the task of scheduling and executing the product supply service. The order holon performs scheduling using off-line optimization methods, so it will need the atomic resources holon information and the store holons information, obtained by holons negotiation. A second task of the order holon is based on the supervision of real-time scheduling; if any disturbance occurs in the execution of the same, the holon order must take corrective measures with the new information and perform a rescheduling. This procedure will be discussed in the next section.

On the other hand the refinery resource holon is composed of tank holons, which for the case under study is considered an inexhaustible supply. Therefore, its

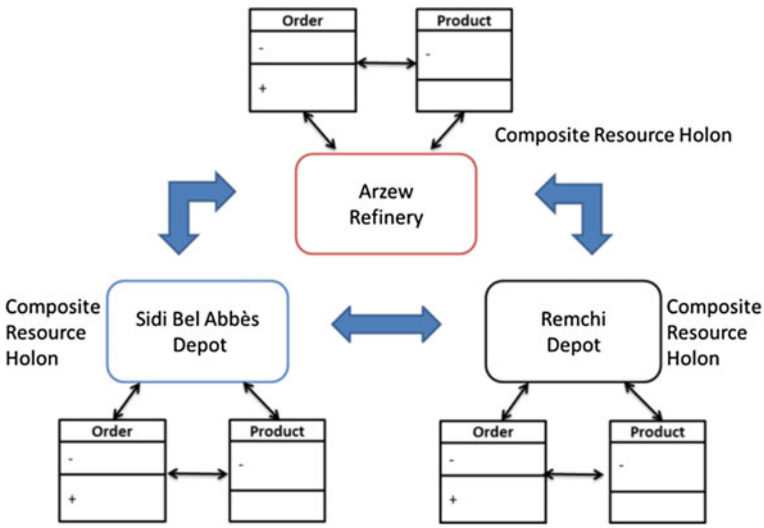


Fig. 5 Distribution level holonic architecture

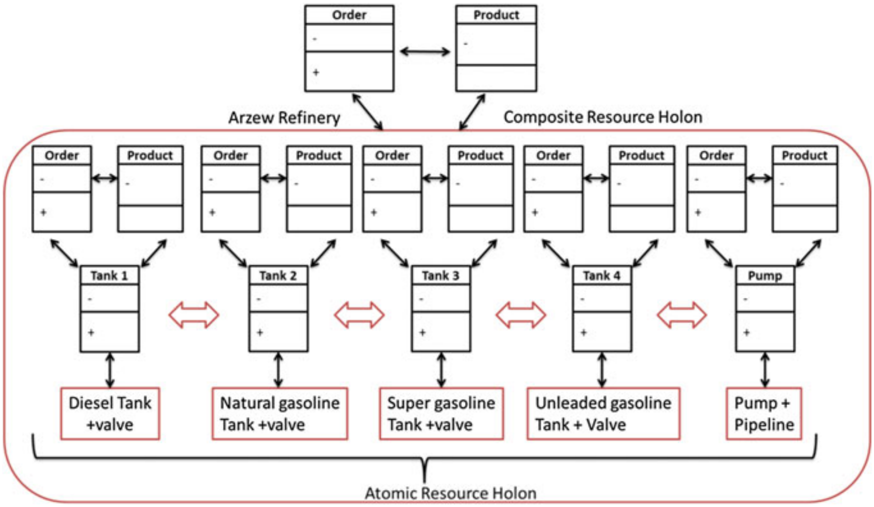


Fig. 6 Refinery composite holon

function is to switch the supply valve depending on the product to be supplied. If there is no inexhaustible supply, this holon should supply the level status of each product to be considered during the scheduling. The other holon available in the refinery holon is the pipeline holon; its task is based on determining the availability of the pipeline as well as providing the products density measure service and the control pumps service during product supply.

The other two holons represented in Fig. 5 are the products depots holons of Sidi Bel Abbès and Remchi. Both composite holons are made up of diesel tanks, super gasoline tank, and normal gasoline tank; this latter depot additionally has unleaded gasoline tank and contaminated product tank. Both depots offer the service of storage of finished product (normal gasoline, super gasoline and diesel). In addition, Remchi depot offers storage service of unleaded gasoline and contaminated product, Fig. 7.

Each holon depot has also associated a product holon and an order holon. The information possessed by the product holon is the characteristic of each product to be received; among them we must highlight the product's density since this information is used to monitor the products in the pipeline. The order holon contains information on the capacity of each tank resource holon. The method used to obtain this information from the compound holons is detailed in [5, 11]. The service provided by the order holon is based on online monitoring of the product tanks filling. If there is a change in the demand, the order holon establishes a new negotiation with the refinery holon to obtain a corrective action.

This description of the adaptation of H²CM reference architecture to this new case study outlines the adaptability of the architecture which was designed generic enough to cope with multiple type of HDS.

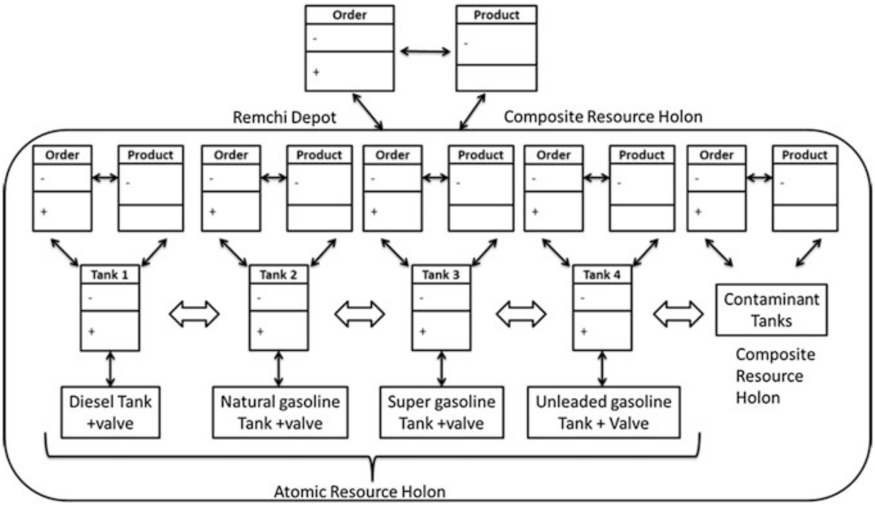


Fig. 7 Depot composite holon

4 Algorithms

The main goal of the product supply scheduling algorithm is to minimize the amount of contaminated product while supplying the tanks to ensure the demand. For this, it is necessary to execute an optimization algorithm that guarantees the desired objective.

Usual optimization methods have high calculation times which make them difficult to implement online. Thus, the system can perform a first schedule using optimal methods since it has sufficient time to perform the calculation, but this schedule needs to be updated in real time using other techniques. This reschedule is obtained by negotiating between the refinery holon and the depot holons.

Once the schedule is calculated (by any method), the scheduling supervision process begins. If an unforeseen event occurs in the product supply process, the order holon needs to make a decision based on the information obtained through the negotiation of the holons in order to solve this new disturbance. It is obviously difficult to exhaustively list all the disturbances that may occur in the system, but the following ones can be considered as the most frequent and impacting ones:

1. Variation in demand for products depots
2. Variation of the tanks' capacity of contaminated product
3. Decreased pump performance of the pipeline system.

These disturbances, when occurring on a system controlled by a schedule based on optimal methods without adjustments in line, can cause changes in the system that result in a shutdown of the functions in the transport system. For this, an online

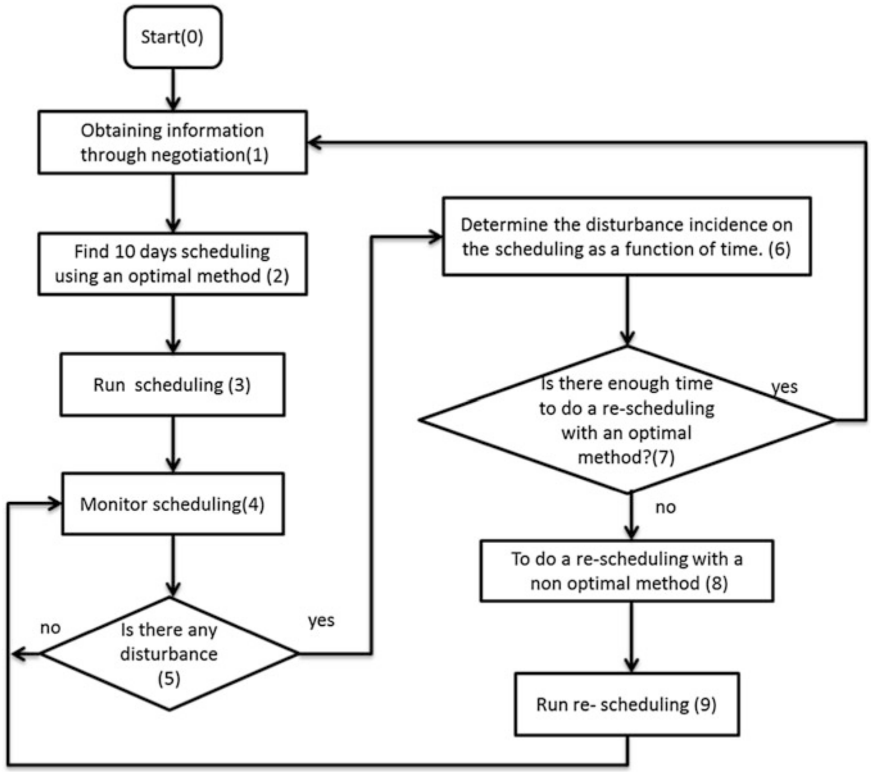


Fig. 8 Scheduling and re-scheduling algorithm

scheduling algorithm is proposed, that aims to find a solution to the presented perturbation, see Fig. 8.

In Fig. 8 on can observe that once the disturbance is present, the incidence on the current planning is calculated as a function of time. If the present disturbance is not so aggressive and immediate changes are not needed, then rescheduling will be performed by optimal methods; otherwise rescheduling will be done by any other computational method that is fast enough to find a solution. An online schedule algorithm for tanks filling minimizing the number of switches of the unique server was proposed in [8], and might be adapted to this specific case study. The main idea is to try and maximize the duration the server remains in the same state, while avoiding the situation where the rest of tank are empty at the same time. Therefore, the algorithm tries to anticipate the reservation of the server if the time slot where some tank would be empty is already scheduled to another tank. A time lapse is also defined. This time lapse represents the length of the reservation a tank tries to schedule. If no solution can be found by the system with the predetermined time lapse, then the algorithm loops with a shorter time lapse. The calculation time of the algorithm is very short, which makes us believe that this kind of scheduling, although it does not guarantee the best minimization of the amount of generated contaminated product, is a good candidate for a pertinent control in the situations where the input data are frequently disturbed.

The algorithms that need to be developed in this case study greatly differ from the ones previously published [8]. Indeed, the former was related to a single server of water and no setup times, while this is related to a single server generating setup wastes denoted contaminated product in this description.

5 Conclusion

Pipeline transport systems are complex HDS that can be scheduled by optimal methods, but they have little flexibility facing disturbances, which makes them difficult to control using online methods. To this end, the H^2CM implementation on HDS was previously proposed [5] in order to give flexibility to those systems.

To demonstrate the application of H^2CM on HDS, a case study of the multi-product ASR pipeline was used in this article. This preliminary work outlined the adaptation of H^2CM to this specific case study and showed it was suitable for modelling. Some elements were given about the definition of the future algorithms that will control the whole architecture, with the objective of finding solutions through optimal and non-optimal methods, and also allowing online monitoring and re-scheduling of the system in presence of any deviation.

The perspectives of this preliminary work deal with the actual coding of the architecture and the performance evaluation compared to the scheduling techniques with a perfectly known demand in order to evaluate the optimality of the holonic control. Then, based on this initial analysis, a study will be performed in order to evaluate the robustness of the control in case of demand variation. The H^2CM based control is meant to absorb those uncertainties in real time, which needs to be solved by specific procedures.

References

1. Pettersson, S., Lennartson, B. et al.: Hybrid modelling focused on hybrid Petri nets. In: 2nd Eur. Workshop Real-Time Hybrid Syst. 303–309 (1995)
2. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theor. Comput. Sci. **138**, 3–34 (1995)
3. David, R., Alla, H.: On hybrid Petri nets. Discret. Event Dyn. Syst. **11**, 9–40 (2001)
4. Chokshi, N., McFarlane, D.: A distributed architecture for reconfigurable control of continuous process operations. J. Intell. Manuf. **19**, 215–232 (2008)
5. Indriago, C., Cardin, O., Rakoto, N., Castagna, P., Chacòn, E.: H2CM: A holonic architecture for flexible hybrid control systems. Comput. Ind. **77**, 15–28 (2016)
6. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.: Reference architecture for holonic manufacturing systems: PROSA. Comput. Ind. **37**, 255–274 (1998)

7. Bennacer, D., Saim, R., Abboudi, S., Benameur, B., et al.: Interface calculation method improves multiproduct transport. Oil Gas J. **114**, 74 (2016)
8. Indriago, C., Cardin, O., Bellenguez-Morineau, O., Rakoto, N., Castagna, P., Chacòn, E.: Performance evaluation of holonic control of a switch arrival system. Concur. Eng. (2016). http://doi.org/1063293X16643568
9. Gamboa Quintanilla, F., Cardin, O., Castagna, P.: Product specification for flexible workflow orchestrations in service oriented holonic manufacturing systems. In: Service Orientation in Holonic and Multi-agent Manufacturing and Robotics, pp. 177–193. Springer (2014)
10. Gamboa Quintanilla, F., Cardin, O., L'Anton, A., Castagna, P.: A Petri net-based methodology to increase flexibility in service-oriented holonic manufacturing systems. Comput. Ind. **76**, 53–68 (2016)
11. Indriago, C., Cardin, O., Rakoto, N., Chacón, E., Castagna, P.: Application of holonic paradigm to hybrid processes: case of a water treatment process. Proceedings of SOHOMA'14 Workshop on Service Orientation in Holonic and Multi-agent Manufacturing. Springer (2014)

An Experimental Device to Supervise the Daily Evaporation in the Guérande's Salt Marshes

Antoine Piel, Xavier Huet, Christophe Leray, Benoit Pedrotti, Benoît Parrein and Rémi Lehn

Abstract This article presents a custom-made device designed to monitor water levels in salt marshes. The device is autonomous and embeds a distance sensor to periodically send measurements of water height to a base station using an Ultra High Frequency (UHF) radio link. These are then collected, processed and displayed to salt workers to help them to supervise the salt marshes productivity. An experimental deployment is carried out in the Guérande salt marshes (near the city of Nantes, France). Preliminary results clearly show the water evaporation inside the basin at the end of the afternoon. The device can be easily extended to numerous sensors in order to monitor several sites simultaneously and to control the water entrance through the hatch. More than a technical depiction, this paper describes the requirements of an ancestral task where a simple cyber-physical system (CPS) can efficiently assist humans in their daily activity.

Keywords Guérande's salt marshes · Smart farms · Cyber-physical systems (CPS) · Ultrasonic sensor network

1 Introduction

In the peninsula of Guérande, the salt marshes have been exploited for more than 1500 years. This marsh belongs to the Coastal Plain Subgroups from the Eastern Atlantic Group [1] (one hour drive from the city of Nantes, France). Salt harvesting is done by hand which respects the environment. The trade of salt-worker requires

A. Piel · X. Huet · C. Leray · B. Pedrotti
Polytech Nantes, Nantes, France

B. Parrein
Networks for IoT, LS2N, Polytech Nantes, Nantes, France

R. Lehn (✉)
LS2N, Polytech Nantes, Nantes, France
e-mail: remi.lehn@polytech.univ-nantes.fr

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_32

a particular and regular attention of the level of water contained in a salt marsh, to control evaporation and optimize daily salt production.

The salt-worker would like to have a technological tool allowing to limit his daily controls at his disposal. Another need is to collect and archive the water levels, to build a history of data, to allow knowledge transmission to the younger generation by means of a transmission medium. For this purpose, we propose a prototype of a connected object using a Raspberry PI, an ultrasound distance sensor and an Ultra High Frequency (UHF) serial radio link. It is a preliminary work before an extension to a real sensor network connected to a remote Cloud. Moreover, it is a good opportunity to talk about the marvellous passion of salt harvesting done by very experienced people for many centuries.

The paper is composed of 4 sections. Section 2 introduces the work of salt harvesting and explains the requirements of the salt worker. Section 3 details the experimental device including the sensor and the communication module. Section 4 provides the experiments done in the salt marsh with early results.

2 The Salt Marshes of Guérande

Less than 70 Km from west of Nantes, there is large wetland area where man and ocean have created a unique and protected natural site. The salt marshes extend over more than 2,000 hectares and 9 towns (Fig. 1). Without mechanized processes and for centuries, salt workers use tides and natural energy, and an open-air gingham of ponds to harvest a natural product: the Guérande salt.

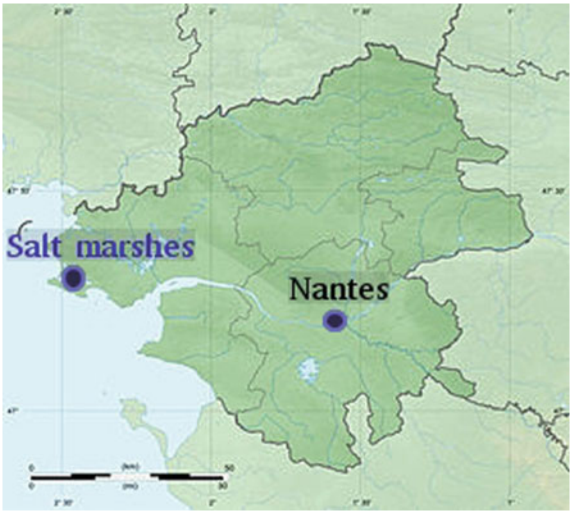


Fig. 1 Localization of the Guérande's salt marsh west from the city of Nantes, France

Fig. 2 General view of a salt marsh. inspired and extracted from [2]

2.1 The Salt Harvesting

Winter and springtime are first reserved to the marshes upkeep and then to its preparation for the summer harvest. During summer, as soon as water is needed, the salt marsh worker opens the hatch during high tides and lets the seawater flood the *decantation basin* (as you can see at point 2 on Fig. 2) which is the first evaporation zone in the circuit that serves as a reserve; here the salt concentration is around 25 gl^{-1}. Then he carefully floods a network of channels and ponds, with the help of gravity. In the last basins, the wind and the sun lead to water evaporation and in the eyelets water reaches high enough levels of concentration for the salt to crystallize (between 250 gl^{-1} and 280 gl^{-1}). In the late afternoon, the salt marsh worker can harvest grains of salt from the eyelets (which are localized as point 9 on Fig. 2) with a wide wooden rake. He produces two types of salt:

- the coarse sea salt, collected on the clay;
- the flower of salt, a fine layer of salt crystals on the water surface.

This last salt keeps its white color because it is not in contact with the clay. The production is much smaller (and much more complicated) than for the coarse sea salt. The price of the flower salt justifies an accurate monitoring of the basins (Fig. 3).

Fig. 3 A salt-worker collecting his material. The small heaps of salt are located on the eyelets

2.2 Requirements of the Salt Marsh Worker

The salt-worker needs to stabilize the water level in the evaporation pond (point 8 in Fig. 2) located just before the eyelets (point 9). It is this basin that determines the water level in the eyelets to allow salt crystals to form during the day. The level of evaporation is set to about 4 cm but it varies during the day depending on the weather conditions. On a sunny day, it can decrease by 3 to 4 cm and in rainy weather, it can rise by more than 5 cm. The salt-worker must therefore know the level of water a range of 0.5 cm to act on the feeding flow of the basin so that the level always remains between 3 and 5 cm. An experienced salt-worker will check the water level on his harvesting sites only twice a day (usually up to 5 different sites): at 8 am and 5 pm. But a trainee or a newly qualified salt-worker needs to do this several times a day to correctly adjust the feed rate and adjust the water level to the nearest millimeter. Therefore it is interesting to propose an automatic water level measurement system. In rainy weather, the system also measures the level of precipitation, which is also very important for salt-workers. A single measurement system per site is sufficient, placed on an evaporation pond acting as a control for this site. To be even more accurate, one or two additional systems could be added per site. After discussion with the Guérande salt-workers interested in the project, it is reasonable to propose a system for a total cost not exceeding 150 €. The complete system must provide a precise measurement up to one millimeter, at a frequency of every 15 min, over the 4 months of the salt harvesting period (June, July, August and September). Data must

Table 1 Salt-worker's functional requirements

Feature	Value
Standard level of evaporation	4 cm
Acuracy of the sensor	0.5 cm
Sensor/site number	5–10
Measurement's period	15 min
Duration of the measurement	4 months
Autonomy of the system	15 days
Device surface	30 cm × 30 cm
maximal total cost	150 €

be archived throughout the harvesting period to allow for the creation of follow-up tables on a day, week or month basis. During training, it is interesting to propose comparative tables between the water level in the evaporation pond and the salinity rate of the evaporation pond. Currently, salinity is measured manually but in the future it would be very beneficial to integrate this measurement in the system. The system must operate with an autonomous 5 V battery for one month (or at least 15 days). It must also blend in the landscape, be positioned above the evaporation pond and be easily accessible. The salty environment is very corrosive for the equipment. Thus, it is recommended to protect the installation with a box (30 cm × 30 cm) which would be placed above the surface of the water (like a nano oil rig). All those requirements are summarized in Table 1.

3 The Experimental Device to Measure the Water Level

For the first prototype, we decide to use a Raspberry Pi as platform for our sensor. To measure the water level, we studied several solutions:

- **Hydrostatic level measurements**, done by float level sensors, pressure sensors or bubbler level sensors. They are simple and not expensive but not precise enough and dependent on the water density. So they are not usable for our project because density varies with salt concentration.
- **Electrical level measurements**, done by conductive or capacitive probes. In our environment the electrical conductivity varies, so we have ruled out this option.
- **Radiation measurements**, done by radar, ultrasonic sensors or optical sensors. Radar sensors are too expensive. Ultrasonic sensors are cheap, accurate enough for short distances (millimeter) and adapted to the environment so we choose this solution.

The selected sensor is the ultrasonic ranging module HC-SR04 from Cytron technologies [3]. It provides 2–400 cm non-contact measurement function. The ranging

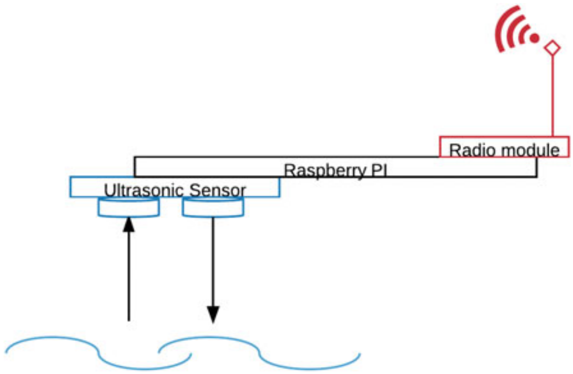


Fig. 4 Complete scheme of the experimental device (communication module included)

accuracy can reach 3 mm, which meets the salt worker's requirements. The module includes ultrasonic transmitters, receiver and control circuit.

To drive the sensor, the Raspberry Pi computer is added as illustrated in Fig. 4 and in the picture of Fig. 5 including a long range and low power radio module. In Fig. 5, we can view the ultrasonic sensor on the left, the Raspberry Pi in the middle, and the USB radio transmitter (RF 434 Mhz) on the right. The input pin of Raspberry Pi GPIO is rated at 3.3 V and the transmitter output is of 5 V so we use fitting resistors of 330 ohms and 470 ohms respectively to bring down the voltage to 3.3 V as exemplified in [4].

The Python GPIO library is necessary for the Rasberry PI [5]. Then, we have developed a Python script to produce measurement, but using C language is advisable to perform automatic and regular measurements with the ultrasonic sensor for greater accuracy.

The transmitter sends an echo which is reflected by the water surface and then retrieved by the receiver. The time difference between transmission and reception of the signal is calculated. With the help of the sonic speed, it is straightforward to get the distance covered by the ultrasound from the sensor to the water level. All measurements are collected by a central platform (the sink) that proposes graphs, history, sensors localisation, alerts and save records. In order to install the system in a salt marsh, we need to protect the embedded electronics as well as the energy

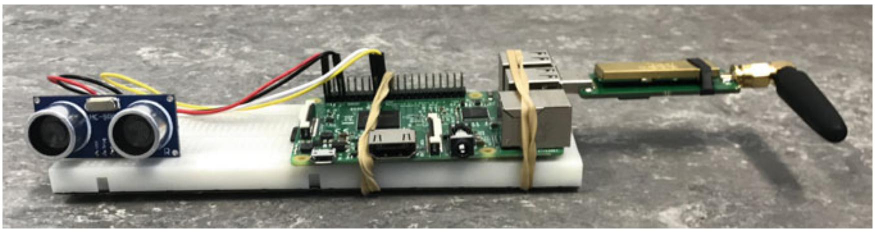


Fig. 5 Zoom on the experimental device

source. As the system will be put in a high humidity environment it may be necessary to place desiccants around the sensor to protect it and improve the product lifetime.

In order to transmit the measurements, we have equipped the Raspberry Pi with a USB radio transmitter (RF) with the technology operating in the 434 MHz bands. The Connect2Pi USB dongle [6] provides a simple wireless bridge and a Linux PC which supports USB serial communications. With a simple script shell of 2 lines, water level measurements, date and time can be sent to the sink for information archiving, processing and remote display G.

4 Experiments and Results

The system was installed in the corner of an area called "evaporation pond", at the edge of the embankment, to be sheltered from wind and rain from the west (ocean side). It is the water level of the "evaporation pond" which acts as a reference value to the salt-worker. This area, by opening or closing hatches made of slates or wooden plates, allows the water level to be regulated in other areas of the salt marsh (see satellite view on Fig. 6).

The system is partly placed on the ground but also attached to a post in the "evaporation pond", above the water (Fig. 7). Having two anchor points ensures stability of the installation and avoids wind disturbances (inconsistent measurements). The measurements were carried out every second during one hour (at the end of the afternoon) within millimeter precision. An average value calculation was then implemented to adjust the different measurements obtained. The statistical characteristics of the measurements are (Fig. 8):

- real value: 25.4 cm
- average value: 25.4 cm
- min value: 25.1 cm
- max value: 27.9 cm.

Fig. 6 Satellite view of the sensor location

Fig. 7 Physical installation of the measuring system in the "evaporation pond"

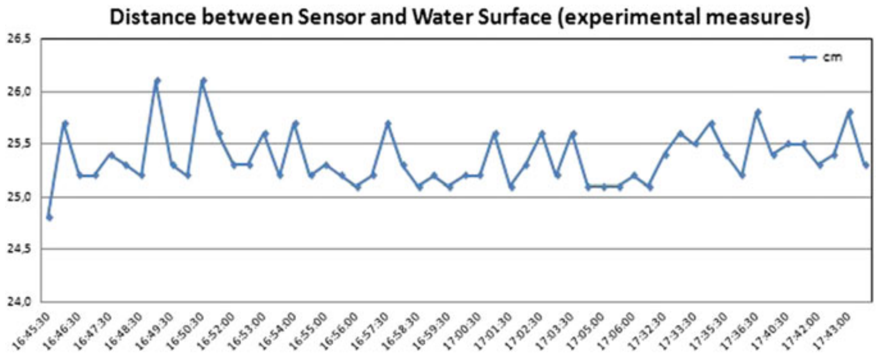


Fig. 8 Graph of experimental raw results

The measurements are filtered using a simple low pass filter to limit the noise due to both the little swash on the water surface caused by the wind as well as some electromagnetic noise. This filter is a classical first order filter:

$$h_t = a \cdot x_t + (1 - a) \cdot h_{t-1}$$

x_t is a measurement at time t and h_t is the filtered value; a is a parameter to control the filter's behaviour: the filter effect are more important when a is close to 0. After filtering and averaging, a value was obtained for every 30 s. With this data, a graph was drawn to allow the salt-worker to quickly visualize the water level of the salt marshes at the location of the probe.

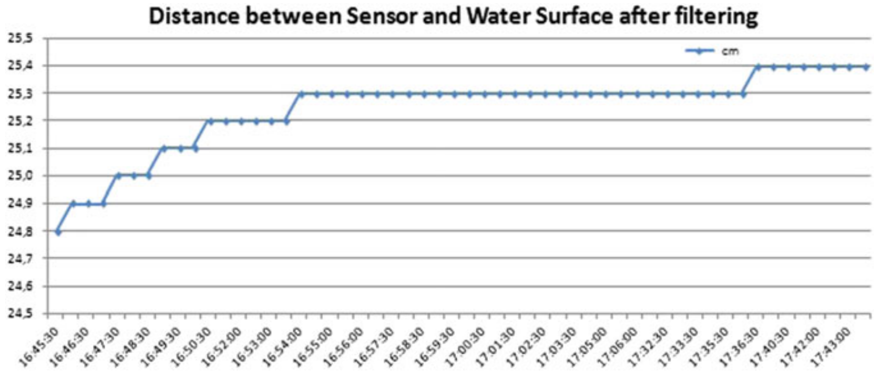


Fig. 9 Graph of experiments results with low pass filter

Raw results are given on Fig. 8. The filtered results are given on Fig. 9. In the second graph, we can clearly see the evaporation effect which is translated there by an increasing distance from the water surface to the ultrasonic sensor.

Thanks to the radio module, we could receive information over 171 m, while the technical characteristics announced a distance of 200 m. which is not sufficient for production. Another radio module or another radio link strategy will be used in a foreseeable future.

5 Future Work

The work presented in this article is a first step in successfully validating the use of an ultrasonic sensor to measure water levels in salt marshes and the feasibility of an UHF data link to push the level values to a ground station.

While this prototype fully meets the first motivation of this work, it would get a full approval at the salt marshes workers' point of view, as soon as

1. The sensors and the emitters are integrated in water and weather resistant boxes that blend discreetly into the salt marshes environment,[1]
2. Radio links are robust and with sufficient range,
3. Data can be transparently collected from several sensors to one to many ground stations,
4. Data can be easily visualized, including precise individual data, summaries and high level informations such as alerts or statistics.

[1] for both aesthetic reasons and to avoid theft…

5.1 Environment Integration

The first two points seem to be the most easily to meet. The use of large PVC tubes that are the same color as the dominant color of salt marshes (light gray) should ensure a good water tightness and airtightness while being sufficiently neutral to radio signals propagation.

5.2 Robust Radio Data Links

The use of professional radio serial links is recommended instead of the cheap and low power ones we used for the prototype. French regulations allow the use of the 433 MHz band (the one we chose for the prototype) without requiring a special license but a full European compliance (`ERC/REC 70-03` recommendation) limit the equivalent isotropically radiated power (e.g. apparent power) of the emitter to 10 mW which limits the range of radio links to 100–200 m in optimal conditions, i.e. direct visibility of emitter to receiver, antennas distant from ground and obstacles, operation in clear weather.

We should investigate the use of other frequencies, such as 868 MHz, 2.4 GHz or 5.5 GHz or 5.8 GHz (which are freely usable with limits of respectively 500 mW, 100 mW, 1W and 50 mW allowing more comfortable ranges up to some kilometers) with more expensive or specific radio serial links. The 868 MHz band seems to be a good candidate with some current investigation in France for its use for IoT advanced projects [7].

The study *IEEE 802.11 a,b,g,n* wireless LAN equipments is a solution we will study as well for data collection from sensors to ground stations. We will investigate both the use of long range wireless bridge with directional antennas in a centralized mode and in ad-hoc mode. In centralized mode one specific radio link will be built between each sensor to one ground station. As a mesh network of sensors in ad-hoc mode, each sensor will be given the capability of forwarding data collected from sensors nodes in a close neighbourhood to other sensors nodes. The former is simpler in design, but the latter should allow using more IEEE 802.11 equipment, and reduce the need to play with the local regulation limits on equivalent isotropically radiated power.

5.3 Data Collection with Multiple Sensors to Multiple Stations

The data model for our prototype is trivial and can be reduced to the acquirement and the collection of {*timestamp*, *water level*} tuples. Of course, the management of distributed data coming from several sensors, distributed to several stations and shared by several salt marshes' workers, requiring several views, including specific and individual data concerning a few evaporation pond as well as aggregated views of data coming from a wider selection of sensors are necessary. An example of such an aggregate view is the average water level in one day with regards to the day before or its derivative.[2]

At a static point of view, the minimum information to maintain is then {*timestamp, water level, sensor, aggregate function*}.

Because of the distribution of sensors and the multiple stations, we currently focus on data models suitable for distributed data and data aggregation. Document or collection oriented data models (e.g. *NoSQL*) that support *map reduce* operators seem to be a suitable choice, with data collection from sensors as the *reduce* function and the computation of aggregate functions as the *map* one [8].

Another benefit of such a data model is the possibility of distributing the computation of aggregate functions on "*smart nodes*" under the hypothesis of ad-hoc links, reducing the amount of data distributed on links to the stations. For example there's no need to transmit every level every minute for a bunch of sensors close to each others if the need is to compute the standard deviation between levels in this bunch of sensors. It's more efficient to elect local smart nodes collecting data and transmitting aggregated values [8].

5.4 High Level Data and Knowledge

Higher level data are needed to allow informed actions from the salt marshes' workers. From their point of view, the system is here both for the on-day maintenance of the salt marshes and for the construction of a knowledge base describing, documenting and supporting their activities.

Aggregate functions and statistics are important and classical aspects for this objective. But we have to investigate the definition of rules or triggers stating predicable events (e.g. sudden rain). This can be included as a special case of aggregate functions, resulting in events or alarms.

The use of such high level indicators raise the problems of having a convenient interface for users who are domain experts but with various skills for IT and devices. For instance, the filtering discussed Sect. 4 only makes sense when filtering coefficients are adjusted with a domain specific objective. While the objective can be clear at the users' point of view, the individual tweaking of such a parameter is not. A solution could be to include very interactive user interfaces with possibilities of simulations, that could help the salt workers to understand the role of each parameter.

[2] how much is the water raising or falling in a given pond?

6 Conclusion

The concept of measurement via an ultrasonic sensor and the prototype implemented in this article were presented and tested together with the salt-worker. He has validated the system allowing it to simply and quickly monitor the water level to optimize his salt production. The system, of moderate cost, is a good tool allowing the transmission of the knowledge for apprentices. As future works we will consider the multiplication of the sensors to distribute the real time monitoring over multiple basins in order to control and regulate automatically the opening and closing of the hatch.

Acknowledgements The authors would like to thank Thierry Giraud, salt marsh worker in Guérande for the his explanations about his expertise and his daily activity. The authors would also like to thank Yann Mahé from the IETR Lab at Polytech Nantes for his help designing the electronic card of the sensor.

References

1. Chapman, V.J.: Salt Marshes and Salt Deserts of the World, Ecology of Halophytes. Academic Press Inc., London, New York (1960)
2. Cap Atlantique: Discover and respect the SALT-MARSHES of the Guerande peninsula (2009). http://www.cap-atlantique.fr/node/935
3. Cytron Technologies, Product Users Manual HC-SR04 Ultrasonic Sensor (2016)
4. Manuel Esteban Blog: Lire un capteur ultrason HC-SR04 avec un Raspberry Pi (2013). http://www.manuel-esteban.com/lire-un-capteur-ultrason-hc-sr04-avec-un-raspberry-pi/
5. Tutorials Raspberrypi.com: Using a Raspberry Pi distance sensor (ultrasonic sensor HC-SR04) (2016). https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/
6. LPRS: eRA Connect2Pi Pi-Compatible RF USB Dongle (2017). http://www.lprs.co.uk/development-tools/evaluation-tools/easyradio-usb-rf-dongles/eraconnect2pi-rf-dongle.html
7. *Nouvelles opportunités pour l'utilisation des bandes 862–870 MHz, 870–876 MHz et 915–921 MHz, consultation publique, 3 juin 2016 - 18 juillet 2016*, *Autorité de régulation des communications électroniques et des postes (ARCEP), Agence nationale des fréquences (ANFR)* ISSN 2258-3106. http://www.anfr.fr/fileadmin/mediatheque/documents/Communiques_Presse/CP_ARCEP_ANFR_2016-06-02_VF2.pdf
8. Ullman, J.D.: Algorithms, designing good mapreduce. XRDS, ACM **19**(1) (2012). ISSN 1528–4972

An Iterative Closest Point Method for Measuring the Level of Similarity of 3D Log Scans in Wood Industry

Cyrine Selma, Hind Bril El Haouzi, Philippe Thomas, Jonathan Gaudreault and Michael Morin

Abstract In the Canadian's lumber industry, simulators are used to predict the lumbers resulting from the sawing of a log at a given sawmill. Giving a log or several logs' 3D scans as input, simulators perform a real-time job to predict the lumbers. These simulators, however, tend to be slow at processing large volumes of wood. We thus explore an alternative approximation techniques based on the Iterative Closest Point (ICP) algorithm to identify the already processed log to which an unseen log resembles the most. The main benefit of the ICP approach is that it can easily handle 3D scans with a variable number of points. We compare this ICP-based nearest neighbour predictor, to predictors built using machine learning algorithms such as the K-nearest-neighbour (kNN) and Random Forest (RF). The implemented ICP-based predictor enabled us to identify key points in

C. Selma (✉) · H. Bril El Haouzi · P. Thomas
Université de Lorraine, Vandœuvre-lès-Nancy Cedex, France
e-mail: cyrine.selma.tn@ieee.org

H. Bril El Haouzi
e-mail: hind.el-haouzi@univ-lorraine.fr

P. Thomas
e-mail: philippe.thomas@univ-lorraine.fr

C. Selma · H. Bril El Haouzi · P. Thomas
CNRS, CRAN, UMR 7039, Vandœuvre-lès-Nancy Cedex, France

J. Gaudreault
Department of Computer Science & Software Engineering, Université Laval, Québec, Canada
e-mail: jonathan.gaudreault@ift.ulaval.ca

J. Gaudreault · M. Morin
FORAC Research Consortium, Université Laval, Québec, Canada
e-mail: mmorin@mie.utoronto.ca

M. Morin
Department of Mechanical & Industrial Engineering, University of Toronto, Toronto, Canada

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_33

using the 3D scans directly for distance calculation. The long-term goal of this on-going research is to integrated ICP distance calculations and machine learning.

Keywords Sawing simulation · Iterative closest point · Machine learning application

1 Introduction

The wood-products industry is facing numerous challenges regarding the maximization of profits and sales when sawmilling. Natural forest resources are characterized by their high heterogeneity and the sawing of a log at a given sawmill is a complex physical process of which the output is not easily forecasted. It is nonetheless critical to decide which contracts to accept based on raw material resources, which sawmill to supply from which cut blocks, and how to configure the parameters of the sawmilling equipment in order to improve the profits and the sales.

Sawing simulators, such as Optitek [1], have been used for years to forecast the production of sawmills and to help at these tasks. They are, for instance, especially useful for sawmill design where time is not an issue [2]. Given the three-dimensional (3D) scan of a log and a sawmill's model, a sawing simulator outputs, among other information, the different pieces of wood that will result from the sawing [1]. Unfortunately, simulators tend to be slow and complex to use when working on large customers' demand under time constraints.

Machine learning algorithms such as K-nearest neighbours (k-NN) [3], decision tree (DT) [4] and random forest (RF) [5] were proposed as a complement to sawing simulation [2]. The authors evaluate machine learning approaches at the task of predicting the lumbers produced at a given sawmill when processing a given log. Based on simple decision rules and using six characteristics of an input log instead of the 3D scan, the predictors built using a machine learning approach are able to approximate the simulator's response of a sawmill producing 19 different lumber products. The technique involved learning the relationship between an input log, described in terms of six features (volume, length, wide-end diameter, narrow-end diameter, and shrinking), and the quantity of every lumber type produced at the plant when sawing it [2].

In this work, we study the usability of the Iterative Closest Point (ICP) algorithm [6, 7] as a mean to measure the resemblance between an input log and already sawn logs. The ICP algorithm has the ability to exploit the entire amount of data (3D scans) available to determine the level of resemblance between two logs by measuring the minimum distance between the two point clouds of their corresponding scans. We used a simple nearest neighbour approach where the produced basket is determined by calculating the minimums of the distances between the input log and the logs of a given training set. Although standard machine learning approaches outperform this simple predictive technique, this research shed some light on the principal noise sources that render the ICP approach less effective for prediction.

This paper is structured as follows. We overview the current approaches used to approximate the sawing process and discuss the possibility of an ICP-based predictor in Sect. 2. Then, in Sect. 3, we explain the ICP method we retained. Experiments and results are presented in Sect. 4. We conclude in Sect. 5.

2 From Sawing Simulation and Machine Learning to Iterative Closest Point

In North America, wood products are standardized by the National Lumber Grades Authority (NLGA) [8]. Each lumber type has specific dimensions, grade, value and price. Lumber production is characterized by a divergent co-productive system; for a single input, it simultaneously produces multiple outputs showing different characteristics. The sawing equipment selects the cutting pattern that maximizes the expected profit. As it is a commodity market, we expect selling the entire production. However, not every mill will produce the same products from the same log. Knowing what each mill would produce from each log (or batch of logs) allows a company to make better decisions thus increasing its efficiency. Sawing simulators has been built especially for that purpose.

Taking the 3D scan of a log as inputs (Fig. 1), a sawing simulator virtually processes the log and allows predicting the products that could be obtained at the modelled sawmill. However, when there are tons of logs that must be processed by

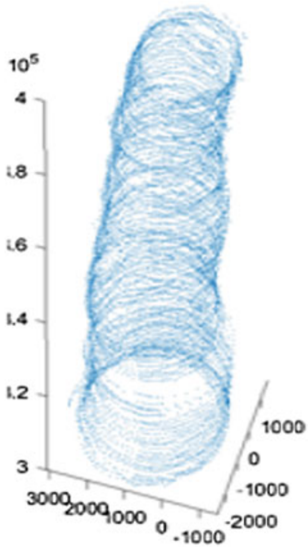


Fig. 1 Input of sawing simulators: 3D scan of a log

sawing simulators in a short period, these simulators tend to be slow and complex to use.

Machine learning was recently proposed as a complement to simulation to overcome this difficulty [2]. The learning algorithm build a predictor from known pairs of input and output where each input is a feature vector representing a log and where its corresponding output are the quantities of each lumber product produced at a given sawmill for that log. Once learned, the predictor can be used to approximate the lumbers resulting from the transformation of an unseen log at that sawmill. The 3D scans, which are made of thousands of points and which are of a variable size, were not used directly for learning.

The ICP-based prediction method we evaluate is based on the nearest neighbour algorithm [3]. The ICP method is used as a measure of distance. Based on the geometrical structure of the logs, the ICP measures the resemblance between the current log and the logs of the training set to find the closest pair. Using an ICP-based nearest neighbour, we calculate the distance between the input log and all logs from the training sets; the minimum of all distances corresponds to the log to which the input log resembles the most. One of the benefits of using ICP as a measure of distance is that it can compare two 3D scans although they might be of a different size in number of points.

3 The Iterative Closest Point Method

The ICP algorithm was introduced by Besl and McKay [6] and Chen and Medioni [7] in the goal of aligning the 3D point clouds of two objects by minimizing the distance between them using geometric transformations (rotations and translations). The ICP algorithm has two steps:

1. The first step consists of determining the correspondence pairs (p, m) from two data sets P and X. The goal is to find for each point p in p its closest point in X.
2. The second step is to apply a transformation (rotation and translation) in order to minimize the distance between the correspondence pairs.

These two steps are repeated until the error is below a given threshold or until the maximum number of iterations is reached.

3.1 ICP Variants

The three main different variants of the ICP found in the literature are the point-to-point method introduced by Besl and McKay [6], the point-to-plane technique by Chen and Medioni [7], and the point-to-projection method by Blais and Levine [9]. We present, in the next section, the point-to-point approach we chose for the experiment.

3.2 The Point-to-Point Registration Method

Let $X = \{\overrightarrow{x_i}\}_{1 \le i \le n_x}$ and $P = \{\overrightarrow{p_j}\}_{1 \le j \le n_p}$ be two shapes. Shape X is the model shape onto which we need to align shape P.

Let $\overrightarrow{q_R} = [q_0\ q_1\ q_2\ q_3]^t$ be a unit quaternion vector, where $q_0 \ge 0$ and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. $\overrightarrow{q_T} = [q_4\ q_5\ q_3]^t$ is the translation vector. $\vec{q} = \langle \overrightarrow{q_R} | \overrightarrow{q_T} \rangle^t$ is the transformation vector. The rotation matrix generated by $\overrightarrow{q_R}$ is given by:

$$R = R(\overrightarrow{q_R}) = \begin{bmatrix} q_0^2 + q_1^2 - (q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 + q_2^2 - (q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 + q_3^2 - (q_1^2 + q_2^2) \end{bmatrix} \quad (1)$$

Let $\vec{p} = (p_1, p_2, p_3)$ be a point of P and $\vec{x} = (x_1, x_2, x_3)$ be a point of X. The Euclidean distance between $\vec{p}$ and $\vec{x}$ is:

$$d(\vec{p}, \vec{x}) = \|\vec{p} - \vec{x}\| = \sqrt{(x_1 - p_1)^2 + (x_2 - p_2)^2 + (x_3 - p_3)^2} \quad (2)$$

The distance between a point $\vec{p}$ and the model shape X is:

$$d(\vec{p}, X) = \min_{1 \le i \le n_x} d(\vec{p}, \overrightarrow{x_i}) \quad (3)$$

The distance between X and the closest point $\overrightarrow{x_j}$ of X is defined by:

$$d(\vec{p}, \overrightarrow{x_j}) = d(\vec{p}, X) = \min_{1 \le i \le n_x} \|\overrightarrow{x_i} - \vec{p}\| \quad (4)$$

For $X = \{\overrightarrow{x_i}\}_{1 \le i \le n_x}$ and $P = \{\overrightarrow{p_j}\}_{1 \le j \le n_p}$ with $n_x = n_p$, the error function to be minimized is given by:

$$f(R, T) = \frac{1}{n_p} \sum_{i=1}^{n_p} \left\| \overrightarrow{x_i} - \left(R\overrightarrow{p_i} + \vec{T} \right) \right\|^2 \quad (5)$$

The goal of this algorithm is to find the optimal transformation (R, T) that minimises $f(R, T)$.

The centres of mass of P and X are given by:

$$\overrightarrow{\mu_p} = \frac{1}{n_p} \sum_{i=1}^{n_p} \overrightarrow{p_i} \tag{6}$$

$$\overrightarrow{\mu_x} = \frac{1}{n_x} \sum_{i=1}^{n_x} \overrightarrow{x_i} \tag{7}$$

The cross-covariance matrix of P and X is:

$$\sum_{px} = \frac{1}{n_p} \sum_{i=1}^{n_p} \left[\left(\overrightarrow{p_i} - \overrightarrow{\mu_p} \right) \left(\overrightarrow{x_i} - \overrightarrow{\mu_x} \right)^t \right] = \frac{1}{n_p} \sum_{i=1}^{n_p} \left[\overrightarrow{p_i}\, \overrightarrow{x_i}^t \right] - \overrightarrow{\mu_p}\, \overrightarrow{\mu_x}^t \tag{8}$$

In order to calculate the vector $\Delta = [H_{23} H_{31} H_{12}]^t$ we use the matrix H_{ij} where: $H_{ij} = \left(\sum_{px} - \sum_{px}^T \right)_{ij}$. Δ is used to calculate $Q\left(\sum_{px} \right)$.

$$Q\left(\sum_{px} \right) = \begin{bmatrix} tr\left(\sum_{px} \right) & \Delta^t \\ \Delta & \sum_{px} + \sum_{px}^t - tr\left(\sum_{px} \right) I_3 \end{bmatrix} \tag{9}$$

where I_3 is the identity matrix.

The optimal rotation vector $\overrightarrow{q_R} = [q_0\ q_1\ q_2\ q_3]^t$ corresponds to the maximum eigenvalue of $Q\left(\sum_{px} \right)$.

The optimal translation vector is:

$$T = \overrightarrow{q_T} = \overrightarrow{\mu_x} - R\overrightarrow{\mu_p} \tag{10}$$

The ICP algorithm is described below:

1. Given the point set P and the model shape X, the first step in the ICP algorithm is to initialize the iteration by $P_0 = P$, $\overrightarrow{q_0} = [1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^t$ and $k = 0$.
2. The following steps are then applied until the convergence within a tolerance τ:
   a. Search the closest points: $Y_k = (P_k, X)$
   b. Compute the registration: $\left(\overrightarrow{q_k}, d_k \right) = Q(P_0, Y_k)$
   c. Apply the registration: $P_{k+1} = \overrightarrow{q_k}(P_0)$
   d. Stop the iteration when the change is mean-square error falls below a present threshold $\tau > 0$ specifying the desired precision of the registration: $d_k - d_{k+1} < \tau$.

ICP moves the shape object in order to best align it with the model shape. Translations and rotations are applied iteratively in order to minimize the error metric. The algorithm has been proved to converge to a local minimum.

4 Experiments

The goal of the experiment is to evaluate a predictive approach based on a standard ICP algorithm at handling the 3D scans for prediction purposes. No filtering of the 3D scan is performed which means that some scans are missing points. The ICP-based predictor is compared to predictors built using the machine learning algorithms presented in [2]. The ICP algorithm used to compute the distance between the scans is a standard algorithm from the MATLAB machine learning library [6].

4.1 Data

A total of 1207 logs were used for the experiments. The training set contains 724 logs (60% of the available datasets) and test set contains 483 logs (40% of the logs). As 736 of the 1207 logs only produce wood chips (they are too small to produce lumber) we also tested for a subset of our dataset from which we removed those (441 logs in the training set, 295 in the testing set) leading to two different datasets. We repeated the experiment 10 times, using different partitions of each dataset into a training and a test set. The sawmill has 19 lumber types leading to an output vector $y \in \mathfrak{R}^{19}$.

For the ICP-based prediction method, the input for the algorithm is the 3D vector containing the points that represent the log on a 3D axis. This vector is variable in length since the number of points in each 3D scans varies.

4.2 Performance Evaluation

We evaluate the performance of the predictors built by each algorithm, including the ICP-based algorithm, using the following metrics [2]: the zero-one loss, the hamming distance, the augmented hamming distance, the prediction score, the production score, and the prediction and production area score.

A zero-one score s^z of 1 indicates that the predicted output basket $\hat{y} \in N^p$ equals the real output basket $y \in N^p$, otherwise s^z is equal to 0 [10]:

$$s^z(\hat{y}, y) = \begin{cases} 1 & if\ \hat{y} = y \\ 0 & otherwise \end{cases} \tag{11}$$

Given an output vector $y \in N^p$ and a predicted output $\hat{y} \in N^p$, the hamming distance d^H represents the average of number of prediction errors across products [10]:

$$d^{H}(\widehat{y}, y) = 1/p \sum_{j=1}^{p} f(\widehat{y}_j, y_j) \tag{12}$$

where:

$$f(\widehat{y}_j, y_j) = \begin{cases} 1 & \mathit{if}\ \widehat{y}_j \neq y_j \\ 0 & \mathit{otherwise} \end{cases} \tag{13}$$

Given an output basket $y \in N^p$ and a predicted output $\widehat{y} \in N^p$, the augmented hamming distance d^{H+} is the sum of the ratios of the minimum between the predicted and the real quantity over the maximum of these quantities averaged across products [2].

$$d^{H+}(\widehat{y}, y) = 1/p \sum_{j=1}^{p} 1 - f(\widehat{y}_j, y_j) \tag{14}$$

where:

$$f(\widehat{y}_j, y_j) = \begin{cases} 1 & \mathit{if}\ \widehat{y}_j = y_j \\ \frac{\min(\widehat{y_j, y_j})}{\max(\widehat{y_j, y_j})} & \mathit{otherwise} \end{cases} \tag{15}$$

Given an output basket $y \in N^p$ exists also in the predicted output $\widehat{y} \in N^p$, the prediction ratio score s^{pre} is the average bounded ratio of the real production on the predicted production [2].

$$s^{pre}(\widehat{y}, y) = 1/p \sum_{j=1}^{p} \min\left(1, \frac{\max(\widehat{y}_j, \varepsilon)}{\max(y_j, \varepsilon)}\right) \tag{16}$$

Given an output basket $y \in N^p$ exists also in the predicted output $\widehat{y} \in N^p$, the production ratio score s^{pro} is the average bounded ratio of the predicted production on the real production [2]:

$$s^{pro}(\widehat{y}, y) = 1/p \sum_{j=1}^{p} \min\left(1, \frac{\max(\widehat{y}_{j_j}, \varepsilon)}{\max(y_{j_j}, \varepsilon)}\right) \tag{17}$$

Given an output basket $y \in N^p$ exists also in the predicted output $\widehat{y} \in N^p$, the production and prediction area score $s^{pro\ x\ pre}$ is computed as the multiplication of the production and the prediction scores [2]:

$$s^{pro\ x\ pre}(\widehat{y}, y) = s^{pro}(\widehat{y}, y)\ s^{pre}(\widehat{y}, y) \tag{18}$$

The ground-truth output y and the predicted output $\hat{y}$ are filtered before computing the scores to avoid overestimating the predictive performance of the evaluated predictor. The filtering consists in removing the product pairs for which both the real and the predicted values are 0.

4.3 Results and Discussion

Using the ICP algorithm, it was possible to determine for each input log from the test set, the training log to which it resembles the most. As a final phase the performance scores were calculated. Table 1 contains the average scores for the ICP-based predictor as well as the average scores obtained by the predictors built using machine learning algorithms (as presented in [2]).

It is worth mentioning that without the filtering of the couples equal to zero during the computation of the scores (see Sect. 4.2), the ICP-based predictor reached more than 90% accuracy with the $s^{pro \times pre}$ score.

When filtering the real and the predicted baskets before computing the scores, the evaluation scores for the ICP-based predictor inevitable goes down. This behaviour was also observed for the standard machine learning predictors [2]. In all cases, the scores of the ICP-based predictor are better than the MEAN algorithm while being worse than the ones obtained by the predictor built using a machine learning approach. This is expected since the ICP-based predictor uses a simple nearest neighbour approach. These results also highlight some of the difficulties encountered by an ICP-based predictor while comparing the 3D scans. We recall that no pre-filtering of the scans is made in this case.

Table 1 Average test scores on 10 runs with random partitions of the data

Without empty baskets							
Scores	MEAN	DT	RF	KRR	KRR-NO	K-NN	ICP
s^z	0.0722	0.5834	0.6088	0.5376	0.5519	0.56	0.1705
$1\text{-}d^H$	0.1298	0.6051	0.6331	0.5809	0.5917	0.5874	0.1879
$1\text{-}d^{H+}$	0.2037	0.7172	0.7423	0.7046	0.7118	0.6989	0.2662
s^{pre}	0.537	0.841	0.8538	0.8402	0.8432	0.7982	0.5988
s^{pro}	0.6666	0.8762	0.8886	0.8644	0.8686	0.9007	0.6674
$s^{pro \times pre}$	0.3109	0.7571	0.7779	0.7426	0.7511	0.7344	0.4071
With empty baskets							
Scores	MEAN	DT	RF	KRR	KRR-NO	K-NN	ICP
s^z	0.1905	0.7006	0.7265	0.6841	0.6919	0.6979	0.3805
$1\text{-}d^H$	0.2102	0.7159	0.7398	0.7077	0.7138	0.7137	0.4037
$1\text{-}d^{H+}$	0.2428	0.7839	0.8044	0.783	0.786	0.7813	0.4798
s^{pre}	0.6332	0.8894	0.8945	0.8885	0.8904	0.8635	0.7775
s^{pro}	0.6096	0.8944	0.9099	0.8946	0.8957	0.9178	0.7022
$s^{pro \times pre}$	0.2964	0.8056	0.8256	0.8044	0.8077	0.8012	0.5427

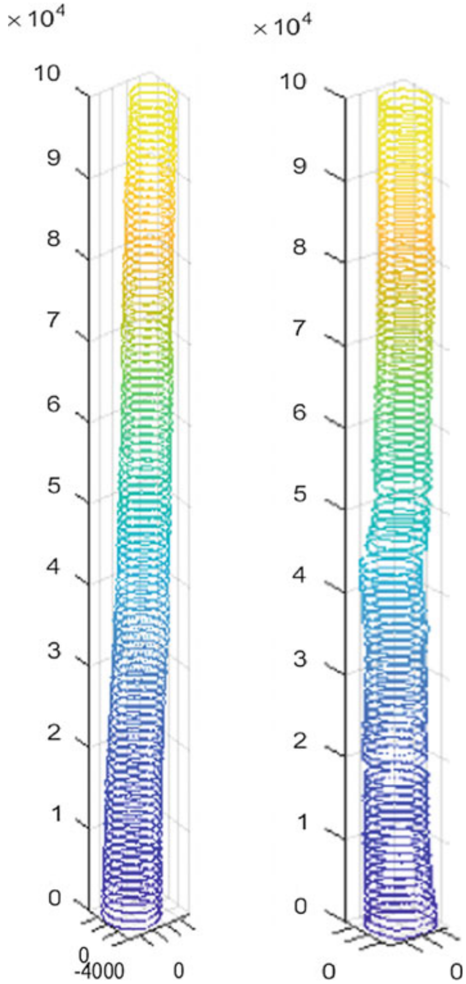


Fig. 2 3D representations of two logs for which the ICP distance is large

The first encountered difficulty is that some logs, although separated by a large ICP distance, share the same basket of products. This is the case of the logs presented in Fig. 2. These logs are separated by a large distance using the ICP algorithm, and they are not considered as similar, even though they have the same basket of products. Our hypothesis is that more data will help in overcoming this issue.

Another difficulty encountered is that some logs that are similar do not share the same basket of products. This is the case for the in the example of Fig. 3. The two logs have a high level of similarity according to the ICP distance, but the first log has an empty basket of products and the second one has a basket that contains two products type 2.

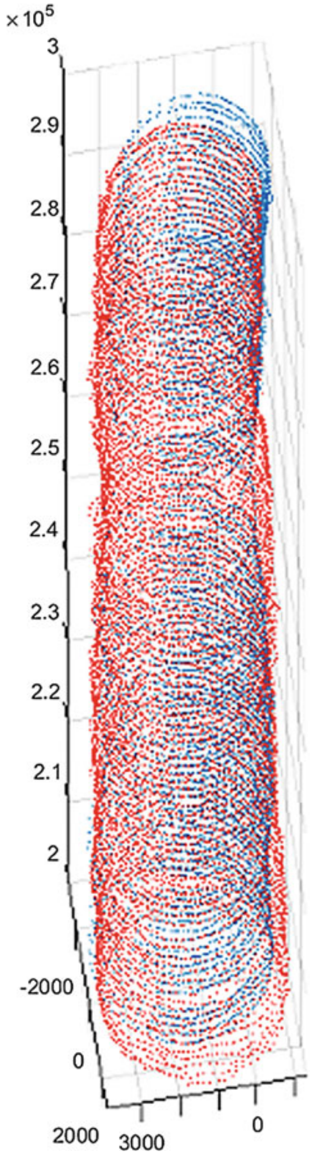


Fig. 3 3D representations of two logs for which the ICP distance is small

5 Conclusion

In this paper, we used the ICP algorithm to compute the distance between pairs of 3D log scans. By minimizing the distance between two given logs, this method is able to determine to which log from a training set an unseen log resembles the most although the 3D scans might differ in the number of points.

The ICP associates a distance to the compared pairs; when this distance is too large, we can understand that the compared couple doesn't have a high level of similarities. This intuition leads to the evaluation of a simple predictor based on the nearest neighbour algorithm.

Coupling the ICP with a machine learning method is one of the perspectives to be considered for the amelioration of the prediction process. As also discussed in the experiments section, we saw that direct use of the logs' 3D scans by an ICP-based predictor might require specific normalization procedures.

References

1. FPInnovations: Optitek 10. User's Manual (2014)
2. Morin, M., Paradis, F., Rolland, A., Wery, J., Gaudreault, J., Laviolette, F.: Machine learning-based metamodels for sawing simulation. In: Proceedings of the 2015 Winter Simulation Conference, pp. 2160–2171 (2015)
3. Fix, E., Hodges, J.: Discriminatory analysis. Nonparametric discrimination: consistency properties. Technical report, DTIC Document (1951)
4. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Chapman and Hall, Belmont, CA (1996)
5. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
6. Besl, P., McKay, N.: A method for registration of 3-D shape. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
7. Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. Image Vis. Comput. **10**(3), 145–155 (1992)
8. NLGA Standard Grading Rules For Canadian Lumber (2014 edn.)
9. Blais, G., Levine, M.: Registering multiview range data to create 3D computer objects. IEEE Trans. Pattern Anal. Mach. Intell. **17**(8), 820–824 (1995)
10. Bakir, G., Hofmann, T., Scholkopf, B., Smola, A.J., Taskar, B., Vishwanathan S.V.N.: Predicting Structured Data. MIT press (2007)

Part VIII
Production and Logistic Control Systems

Generic Routings for ConWip Sizing in a Multi-product Environment

Yann Jaegler, Samir Lamouri, Damien Trentesaux, Patrick Burlat and Anicia Jaegler

Abstract In a vulnerable, uncertain, complex, and ambiguous environment, ConWip provides a sustainable, effective and adaptive production control system for manufacturers. The present paper deals with the key questions related to the implementation of ConWip in a high product mix and/or high routing mix environment. To respond to this challenge, generic routing has to be defined to cover all of the routings of the high product mix. Through four algorithms, this paper studies how to define a representative routing. A numerical sample is derived from industrial data. We implement it in the four algorithms to generate four different generic routings. Then, thanks to Wipsim, an engineering tool used in projects to design and improve ConWip lines, we calculate the optimized ConWip parameters for each of them. Finally, we compare the results and highlight some research avenues.

Keywords ConWip • Production control system • Routing • Complex environment • Management of production • Customer-oriented logistics Cyber-physical manufacturing systems • Industry 4.0

Y. Jaegler · S. Lamouri
Arts et Métiers ParisTech, LAMIH UMR CNRS 8201, Paris, France

D. Trentesaux
LAMIH UMR CNRS 8201, Université de Valenciennes et du Hainaut-Cambrésis, Valenciennes, France

P. Burlat
Société Wipsim sas, St Étienne, France

A. Jaegler (✉)
Center of Excellence in Supply Chain, Kedge Business School, Bordeaux, France
e-mail: anicia.jaegler@kedgebs.com

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_34

1 Introduction

In a **V**ulnerable, **U**ncertain, **C**omplex, and **A**mbiguous (VUCA) environment, a cornerstone of manufacturing effectiveness is the right choice of its Production Control System (PCS). Moreover, in the challenging context of sustainable development, including environmental footprint and energy consumption reduction objectives, PCS must also be designed to control—as much as possible—inventory levels and machine efficiency, while at the same time it must support the needs for high customization and reactive reply to evolving customer demands. In such a context, ConWip is a kind of flexible and efficient PCS supporting multiple products and moving demand. Generally speaking, the main goal of ConWip is to control the total amount of work on the production line by keeping it constant. This idea is rooted in the law that Little [1] defined, commonly known as Little's Law:

$$\text{Work in Process} = \text{Throughput} * \text{Makespan}$$

ConWip uses the work orders produced by a planning system to select the part numbers and the corresponding quantities to produce. It aims to keep WIP level constant on a production line. First, a fixed number of cards is defined. Every order has to be matched with a card giving an "authorization to produce" to be processed. The completion of an order at the end of the line releases the associated card, which then becomes available for a new order. This guideline allows keeping smooth flow on the line with no excess of WIP. Authors of [2] have studied the research lines concerning ConWip. The paths deal with implementation and optimization, environment, comparison methodology and simulation tools and return of experience. If we focus on environment paths, three themes of productive contexts are highlighted. One of them is the mix-model characteristics. The key questions are related to high product mix and/or high routing mix. One major direction could be to define generic routing-based ConWip to solve high mix issues. Since 1992, [3] have suggested defining routings in ConWip systems. They theorize about the link between routings and ConWip sizing. Germs and Riezebos [4] and [5] follow this concept and suggest basing ConWip on the load. Moreover, [6] assumes that his model makes a single part type. He suggests studying a more complex system with a multi-product manufacturing environment. Authors of [7] recommend studying complex manufacturing environments; in [8] it is shown that capturing the complex interaction of multi-product settings is crucial. Finally, the idea is to find the best card count for routings in order to optimize ConWip [9, 10]. Authors of [11] explore the effect of routings on separate product performance.

To answer this challenge, a representative routing has to be defined to cover all the routings of the different products. Through four algorithms, this paper studies how to define a representative routing. A numerical sample is derived from industrial data. The four different algorithms are tested with this sample. The reminder of this paper is organized as follows. The algorithms are detailed in Sect. 2. In Sect. 3 we analyze the results of a numerical sample. Finally, concluding remarks and future research avenues are drawn.

2 Generic Routing Design

2.1 Problem Statement

We know how to size a ConWip (card number and lot sizing) focused on one manufacturing routing. Nevertheless, in practice, a production line manufactures different products with different routings. In this case, we don't know how to optimize the sizing of a ConWip. This means that a practitioner will not be able to calculate the best or even good parameters to size his ConWip. He will use experience or proceed by testing different parameters. In the case of a moving environment, this experience or testing driven optimization will take too much time to stand for a real solution. In this context, the question this paper will help to answer is the following: how can one decide which parameters he will use to size a ConWip system in a moving multi-product environment? To answer this question, we need to simplify the problem by reducing it to a single-product environment. To achieve this problem reduction, we define a representative routing of all the manufacturing routings. This representative routing is called a "generic routing". It will be used as the routing of the single product environment which simulates the multi-product environment we need to work on. Through this problem reduction, it will be possible to dimension the ConWip of the line or the shop directly in the workshop by the practitioners. We search a generic routing that provides the answer to the constraints: no turn back in line for any product routings, compared to the generic routing, to guarantee easy to use visual management. The shorter the generic routing, the better (avoid workstation redundancy).

2.2 Methodology

To define this generic routing, the methodology below is used. We build an algorithm to find a generic routing for a combination of routings. We choose a sample of data (products, routings, and mixed model) from a real case.

2.3 Algorithms

We define four algorithms. The following parameters are used:

W_l	*the workstation number l of the work shop or production line*, $l = 1, \ldots, o$
n	*the number of products*
m	*the number of operations*
i	*the index of the product*, $i = 1, \ldots, n$
j	*the index of the operation*, $j = 1, \ldots, m$
mi	*the number of operations in the routing of product i*

$mIGRO_k$	*the number of operations in the intermedioate generic routing k*
$mGRO$	*the number of operations in the final generic routing*
T_{ij}	*the setup time required to the operation j for the product i*
P_{ij}	*the processing time of operation j on product i*
$O_{ij}\left(W_{ij}, T_{ij}, P_{ij}\right)$	*operation j of the routing of product i*
$W\left(O_{ij}\right)$	*workstation of the operation j of the routing of product i*
β_i	percentage *weight associated to the product i in the mix*

which implies:

$$\sum_{i=1}^{n} \beta_i = 1$$

$IGRT_{kj}$	*the setup time of operation j for the intermediate generic routing k*
$IGRP_{kj}$	*the processing time of operation j for the intermediate generic routing k*
$IGRO_{kj}\left(W_l, IGRT_j, IGRP_j\right)$	*the operation j of the intermediate generic routing k*
GRT_j	*the setup time of operation j for the generic final routing*
GRP_j	*the processing time of operation j for the final generic routing*
$GRO_j\left(W_l, GRT_j, GRP_j\right)$	*the operation j of the final generic routing*

Two by two routings algorithm

This algorithm adds the routings two by two. This means that for two routings, we compare the first operation of the first routing with the first operation of the second routing, and the second operation of the first routing with the second operation of the second routing. We add the operations if they are similar, and if not we keep the first one. We apply this process to all of the operations of the two routings. We obtain a new routing—the first intermediate generic routing. We add this new routing to the 3rd routing to obtain the 2nd intermediate generic routing. The process goes on until all products' routings are added. Once completed, the final generic routing is obtained.

With the notations, we define operations j of the intermediate generic routing k:
For $k \neq 1$:

$$IGRO_{kj} = \left\{ \begin{array}{c} \beta_{k+1} O_{k+1j} + IGRO_{k-1j} + \beta_{k+1} O_{k+1j+1} + IGRO_{k-1j+1} \\ \text{if } W\left(O_{k+1j}\right) = W\left(IGRO_{k-1j}\right) = W\left(O_{k+1j+1}\right) = W\left(IGRO_{k-1j+1}\right) \end{array} \right\} \tag{1}$$

$$IGRO_{kj} = \left\{ \begin{array}{c} \beta_{k+1} O_{k+1j} + IGRO_{k-1j} + \beta_{k+1} O_{k+1j+1} \\ \text{if } W\left(O_{k+1j}\right) = W\left(IGRO_{k-1j}\right) = W\left(O_{k+1j+1}\right), \text{and} <> W\left(IGRO_{k-1j+1}\right) \end{array} \right\} \tag{2}$$

$$IGRO_{kj} = \left\{ \begin{array}{c} \beta_{k+1} O_{k+1j} + IGRO_{k-1j} + IGRO_{k-1j+1} \\ if\ W\left(O_{k+1j}\right) = W\left(IGRO_{k-1j}\right) = W\left(IGRO_{k-1j+1}\right), \text{and} <> W\left(O_{k+1j+1}\right) \end{array} \right\} \tag{3}$$

$$IGRO_{kj} = \left\{ \begin{array}{c} \beta_{k+1} O_{k+1j} + \beta_{k+1} O_{k+1j+1} + IGRO_{k-1j+1} \\ if\ W\left(O_{k+1j}\right) = W\left(O_{k+1j+1}\right) = W\left(IGRO_{k-1j+1}\right), \text{and} <> W\left(IGRO_{k-1j}\right) \end{array} \right\} \tag{4}$$

$$IGRO_{kj} = \left\{ \begin{array}{c} \beta_{k+1} O_{k+1j} + IGRO_{k-1j} \\ if\ W\left(O_{k+1j}\right) = W\left(IGRO_{k-1j}\right), \text{and} <> W\left(O_{k+1j+1}\right), \text{and} <> W\left(IGRO_{k-1j+1}\right) \end{array} \right\} \tag{5}$$

$$IGRO_{kj} = \left\{ \begin{array}{c} \beta_{k+1} O_{k+1j} + \beta_{k+1} O_{k+1j+1} \\ if\ W\left(O_{k+1j}\right) = W\left(O_{k+1j+1}\right), \text{and} <> W\left(IGRO_{k-1j}\right), \text{and} <> W\left(IGRO_{k-1j+1}\right) \end{array} \right\} \tag{6}$$

$$IGRO_{kj} = \left\{ \begin{array}{c} \beta_{k+1} O_{k+1j} + IGRO_{k-1j+1} \\ if\ W\left(O_{k+1j}\right) = W\left(IGRO_{k-1j+1}\right), \text{and} <> W\left(O_{k+1j+1}\right), \text{and} <> W\left(IGRO_{k-1j}\right) \end{array} \right\} \tag{7}$$

$$IGRO_{kj} = \left\{ \begin{array}{c} \beta_{k+1} O_{k+1j} \\ if\ W\left(O_{k+1j}\right) <> W\left(IGRO_{k-1j+1}\right), \text{and} <> W\left(O_{k+1j+1}\right), \text{and} <> W\left(IGRO_{k-1j}\right) \end{array} \right\} \tag{8}$$

For $k = 1$, the process is the same, uses $\beta_1, O_{1j}, \beta_2, O_{2j}$ instead of $\beta_{k+1}, O_{k+1j}, O_{k+1j+1}, IGRO_{k-1}$, $IGRO_{k-1j+1}$ except in cases (4) and (7).

To digitize the algorithm, we use three products with three different operations (see Tables 1 and 2):

Highest occurrence algorithm

We consider all operations on all products. The workstation $(W(GRO_1))$ of the 1st operation (GRO_1) of generic routing is one (in $W\left(O_{1j}\right), j = 1, \ldots, m$) with highest occurrence. We calculate the processing time GRP_1 of the first operation of the generic routing by adding the processing times of O_{1j} which have the workstation $(W(GRO_1))$ multiplied by their weights in the mix. The same calculation is performed for the set up time. If all operations are different, we keep that of the first product. All unused operations are offset to one side. We re-apply the process until last task. For generic routing calculate C_l *the occurences of* $W\left(O_{jl}\right), l = 1, \ldots, o$

Then, we select z such as: $C_z = Max(C_l), l = 1, \ldots, o$, and calculate:

Table 1 Example data (ST: setup time, PT: process time)

Products		1	2	3
Mix weight (%)		33	33	33
Operation 1	name	A	B	A
	ST	0,5	0,5	0,5
	PT	1	3	1
Operation 2	name	B	A	B
	ST	1	0,5	1
	PT	3	3	3
Operation 3	name	C	C	A
	ST	0	0	0,5
	PT	0,5	0,5	4
Operation 4	name			C
	ST			0
	PT			0,5

Table 2 Generic routing for the first algorithm

Products	Operations							
	1		2		3		4	
	A		B		C			
1	0.5	1	1	3	0	0.5		
	B		A		C			
2	0.5	3	0.5	3	0	0.5		
	A + A		B + B		C + C			
New routing	0.3	1.3	0.5	2.0	0.0	0.3		
	A		B		A		C	
3	0.5	1	1	3	0.5	4	0	0.5
	A + A		B + B		C + C		A	
Generic routing	0.5	1.7	0.8	3.0	0.2	0.3	0.2	1.3

$$GRT_j = \left\{ \sum_{i=1}^{n} \beta_i T_{ij} \; with \; i/W\left(O_{ij}\right) = W(O_{iz}) \right\}$$

$$GRP_j = \left\{ \sum_{i=1}^{n} \beta_i P_{ij} \; with \; i/W\left(O_{ij}\right) = W(O_{iz}) \right\}$$

Finally, we define: $GRO_j\left(W\left(O_{jz}\right), GRT_j, GRP_j\right)$. With the previous example, we obtain Table 3.

Highest processing time algorithm

We consider all the first operations of all of the products. We define the workstation $(W(GRO_1))$ of the first operation (GRO_1) of the generic routing as the workstation

Table 3 Generic routing for the second algorithm

Products								
	A		B		C			
1	0,5	1	1	3	0	0,5		
	B		A		C			
2	0,5	3	0,5	3	0	0,5		
	A		B		A		C	
3	0,5	1	1	3	0,5	4	0	0,5
	A + A							
Generic routing	0,3	0,7						
	A		B		C			
1	0,5	1	1	3	0	0,5		
			B		A		C	
2			0,5	3	0,5	3	0	0,5
	A		B		A		C	
3	0,5	1	1	3	0,5	4	0	0,5
	A		B + B + B					
Generic routing	0,3	0,7	0,8	3				
	A		B		C			
1	0,5	1	1	3	0	0,5		
			B		A		C	
2			0,5	3	0,5	3	0	0,5
	A		B		A		C	
3	0,5	1	1	3	0,5	4	0	0,5
	A		B		A + A			
Generic routing	0,3	0,7	0,8	3	0,3	2,3		
	A		B				C	
1	0,5	1	1	3			0	0,5
			B		A		C	
2			0,5	3	0,5	3	0	0,5
	A		B		A		C	
3	0,5	1	1	3	0,5	4	0	0,5
	A		B		A		C + C + C	
Generic routing	0,3	0,7	0,8	3	0,3	2,3	0	0,5

(throughout $W(O_{1j}), j = 1, \ldots, m$) which receives the highest operational time, considering all of the first operations of all of the products. We calculate the processing time GRP_1 of the first operation of the generic routing by adding the processing times of O_{1j} which have workstation $(W(GRO_1))$ multiplied by their weights in the mix. The same calculation is performed for the set up time. If all the operations are different, we keep that of the first product. All of the unused operations are offset to one side. We re-apply the process until the last operation.

To define the generic routing, we search: $z / \sum_{i=1}^{n} P_{iz} = Max\left(\sum_{i=1}^{n}\sum_{j=1}^{m} P_{ij}\right)$
Then we calculate:

$$GRT_j = \sum_{i=1}^{n} \beta_i T_{ij} \; with \; i / W\left(O_{ij}\right) = W\left(O_{iz}\right)$$

$$GRP_j = \sum_{i=1}^{n} \beta_i P_{ij} \; with \; i / W\left(O_{ij}\right) = W\left(O_{iz}\right)$$

Finally, we define: $GRO_j\left(W\left(O_{jz}\right), GRT_j, GRP_j\right)$. With the previous example, we obtain Table 4.

Table 4 Generic routing for the third algorithm

Products										
	A		B		C					
1	0,5	1	1	3	0	0,5				
	B		A		C					
2	0,5	3	0,5	3	0	0,5				
	A		B		A		C			
3	0,5	1	1	3	0,5	4	0	0,5		
	B									
Generic routing	0,2	1,0								
			A		B		C			
1			0,5	1	1	3	0	0,5		
	B		A		C					
2	0,5	3	0,5	3	0	0,5				
			A		B		A		C	
3			0,5	1	1	3	1	4	0	0,5
	B		A + A + A		B + B					
Generic routing	0,2	1,0	0,5	1,7	0,7	2				
			A		B		C			
1			0,5	1	1	3	0	0,5		
	B		A				C			
2	0,5	3	0,5	3			0	0,5		
			A		B		A		C	
3			0,5	1	1	3	1	4	0	0,5
	B		A		B + B		A			
Generic routing	0,2	1,0	0,5	1,7	0,7	2	0	1,3		
			A		B				C	
1			0,5	1	1	3			0	0,5
	B		A						C	
2	0,5	3	0,5	3					0	0,5
			A		B		A		C	
3			0,5	1	1	3	1	4	0	0,5
	B		A		B		A		C + C + C	
Generic routing	0,2	1,0	0,5	1,7	0,7	2	0	1,3	0	0.5

Accordion algorithm

For this algorithm, we consider all the product routings at the same time. The workstation of the first operation of the first product ($W(O_{11})$ is considered. If the workstation of the first operation of the second product $W(O_{21})$ is the same, we leave it in this position. If it is not, we put it into position two, which means (O_{21}) becomes (O_{22}) and (O_{21}) is replaced by an empty operation (no workstation, set up time equal 0 and processing time equal 0).

We repeat this process for the first operations of all the products and we add all of the remaining first operations. The second step consists of going through the same process for the second operations of all the products, and so on, up to the last operation for each product. If needed, an operation of the first product may also be skipped and replaced by an empty operation.

The whole process is defined by the following algorithm:

```
For i = 2 to n
    For j = 1 to mi
                                    If W_{i,j} <> W_{i-1,j} then
            If j < mi-1 then
             For k = 0 to mi-j
                    O_{i,mi-(k-1)} = O_{i,mi-}k And O_{i,j} = (0, 0, 0)
                Next
                                    End If
            If j = mi-1 then
             For k = 0 to mi-j
                    O_{i,mi-(k-1)}) = O_{i,mi-k} And O_{i,j} = (0, 0, 0)
                Next
                            O_{i-1,j+1}=O_{i-1,j} And O_{i-1,j}= (0,0,0)
                                    End If
        End If
    Next
Next
For j = 1 to mGRO
```

$$GRO_j = \sum_{i=1}^{n} O_{ij}$$

Next

With the previous example, we obtain Table 5.

Table 5 Generic routing for the fourth example

Products								
	A		B				C	
1	0,5	2	1	3			0	0,5
			B		A		C	
2			0,5	3	0,5	3	0	0,5
	A		B		A		C	
3	0,5	2	1	3	0,5	4	0	0,5
	A		B		A		C	
Generic routing	0,3	1,3	0,8	3,0	0,3	2,3	0,0	0,5

3 Results

3.1 Numerical Sample

In this sample, we consider a multiple operations production shop. The data are from an industrial manufacturer. According to confidentiality rules, the firm cannot be cited in this work. The operations are machining, milling, and control (Table 6). There is a process time associated with each operation of each product that may differ from one product to another. There is a setup time required for each operation. It is also assumed that the shop uses a ConWip production control strategy.

The shop consists of two machines, three milling workstations and one control workstation.

3.2 Results Obtained

Each generic routing is implemented into the software Wipsim, which is one engineering tool used in projects to design and improve ConWip lines.

The software gives the following numerical results in terms of ConWip optimized parameters (Table 7):

The four algorithms generate the following results in terms of generic routings (Table 8):

Table 6 Routings (ST: setup time, PT: process time)

Products		1	2	3	4	5	6
Mix weight (%)		17	17	17	17	17	17
Operation 1	name	machining 1	machining 2	machining 1	machining 1	machining 1	milling 1
	ST	0,5	0,5	0,5	0,5	0,5	0,5
	PT	2	3	2	3	2	2
Operation 2	name	milling 1	milling 2	milling 1	machining 2	machining 2	machining 2
	ST	1	0,5	1	0,5	1	1
	PT	3	3	3	3	3	3
Operation 3	name	machining 2	milling 3	machining 2	milling 1	milling 2	milling 2
	ST	0,5	1	0,5	1	0,5	0,5
	PT	4	3	4	3	4	4
Operation 4	name	milling 2	control	control	control	milling 3	milling 3
	ST	1	0	0	0	1	1
	PT	4	1	0,5	1	4	4
Operation 5	name	control				control	control
	ST	0				0	0
	PT	0,5				0,5	0,5

Table 7 ConWip parameters according to the four models

Model	ConWip tickets	Lot size
Two by two routings	12–14	2
Highest occurrence	6–8	2
Highest operating time	7–9	2–3
Accordion	12–14	2–3

Table 8 Generic routings according to the four models (ST: setup time, PT: process time)

Model		Two by two routings	Highest occurrence	Highest operating time	Accordion
Operation 1	name	milling 1	machining 1	machining 1	machining 1
	ST	0,1	0,3	0,3	0,3
	PT	0,3	1,5	1,5	1,5
Operation 2	name	machining 1	milling 1	machining 2	milling 1
	ST	0,3	0,4	0,3	0,4
	PT	1,5	1,3	1,5	1,3
Operation 3	name	machining 2	machining 2	milling 1	machining 2
	ST	0,6	0,7	0,6	0,7
	PT	2,7	3,3	1,8	3,3
Operation 4	name	milling 2	milling 2	machining 2	milling 2
	ST	0,2	0,4	0,3	0,4
	PT	1,3	2,5	1,8	2,5
Operation 5	name	milling 3	milling 3	milling 2	milling 3
	ST	0,3	0,5	0,4	0,5
	PT	1,3	1,8	2,5	1,8
Operation 6	name	milling 1	control	milling 3	milling 1
	ST	0,5	0,0	0,5	0,2
	PT	1,5	0,5	1,8	0,5
Operation 7	name	control	milling 1	control	control
	ST	0,0	0,2	0,0	0,0
	PT	0,4	0,5	0,7	0,7
Operation 8	name	milling 2	control		
	ST	0,1	0,0		
	PT	0,5	0,2		
Operation 9	name	machining 2			
	ST	0,1			
	PT	0,7			

(continued)

Table 8 (continued)

Model		Two by two routings	Highest occurrence	Highest operating time	Accordion
Operation 10	name	milling 3			
	ST	0,2			
	PT	0,5			
Operation 11	name	milling 2			
	ST	0,2			
	PT	0,7			
Operation 12	name	control			
	ST	0,0			
	PT	0,3			

4 Discussion and Conclusion

The four algorithms generate four different generic routings. The first one has 12 workstations, the second one has 8, and the last two have 7. Thanks to Wipsim, we obtain the optimized ConWip parameters that correspond to the four routings. All of the algorithms give a lot size of 2. The number of ConWip tickets is variable. The highest occurrence algorithm and the highest processing time algorithm bring about very similar parameters. Then, the two by two routings algorithm and the accordion algorithm have a higher but similar number of tickets.

The present work has some limitations, which could bring about additional lines of research. The mixed model is homogenous. Some other industrial complex environments could be defined, i.e. we could render the mixed model heterogeneous. Moreover, the results could also be compared on different criteria such as mix sensibility or manufacturing sensibility, such as sorting product inside the mix in order to eliminate the products that are atypical, such as being associated with a very low mix weight.

In the next step of this research, the four ConWip parameters will be tested in a simulated workshop to select the best generic routing. This will allow us to fully answer the research question. Algorithm two by two routings and Accordion are asymmetrical, since the choice of the first product may be modified in the final generic routing. Future research could be to define the best way to choose the first product.

References

1. Little, J.D.: A proof for the queuing formula: L = λ W. Oper. Res. **9**, 383–387 (1961)
2. Jaegler, Y., Jaegler, A., Trentesaux, D., Burlat, P., Lamouri, S.: The CONWIP production control system: classification and discussion of current and future research. J. Eur. Systèmes Autom. 1–22 (2017)

3. Spearman, M.L., Zazanis, M.A.: Push and pull production systems: issues and comparisons. Oper. Res. **40**, 521–532 (1992)
4. Germs, R., Riezebos, J.: Workload balancing capability of pull systems in MTO production. Int. J. Prod. Res. **48**, 2345–2360 (2010)
5. Ziengs, N., Riezebos, J., Germs, R.: Placement of effective work-in-progress limits in route-specific unit-based pull systems. Int. J. Prod. Res. **50**, 4358–4371 (2012)
6. Khojasteh-Ghamari, Y.: A performance comparison between kanban and CONWIP controlled assembly systems. J. Intell. Manuf. **20**, 751–760 (2009)
7. Parvin, H., Van Oyen, M.P., Pandelis, D.G., Williams, D.P., Lee, J.: Fixed task zone chaining: worker coordination and zone design for inexpensive cross-training in serial CONWIP lines. IIE Trans. **44**, 894–914 (2012)
8. Satyam, K., Krishnamurthy, A.: Performance analysis of CONWIP systems with batch size constraints. Ann. Oper. Res. **209**, 85–114 (2013)
9. Eng, C.K., Sin, L.K.: CONWIP based control of a semiconductor end of line assembly. Procedia Eng. **53**, 607–615 (2013)
10. Prakash, J., Chin, J.F.: Modified CONWIP systems: a review and classification. Prod. Plan. Control. **26**, 296–307 (2015)
11. Bahaji, N., Kuhl, M.: A simulation study of new multi-objective composite dispatching rules, CONWIP, in semiconductor fabrication. Int. J. Prod. Res. **46**, 3801–3824 (2008)

A New Rescheduling Heuristic for Flexible Job Shop Problem with Machine Disruption

Maroua Nouiri, Abdelghani Bekrar, Abderrazak Jemai, Ahmed Chiheb Ammari and Smail Niar

Abstract In real-world manufacturing systems, schedules are often confronted with uncertainty factors such as random machine breakdown, random process time, random job arrivals or job cancellations. A large number of impromptu disruptions frequently affect the scheduled operations and invalidate the original schedule. There is still the need for rescheduling methods that can work effectively in disruption management. In this work, an algorithm for rescheduling the affected operations in a flexible job shop is presented and its performance, with respect to measures of efficiency and stability, is compared with the Right Shift Rescheduling technique. The proposed method is tested on different benchmark scheduling problems with various disruption scenarios. Experimental results show that the proposed rescheduling method improves the efficiency and stability when compared to Right Shift Rescheduling method.

Keywords Rescheduling · Flexible job shop problem · Stability Efficiency · Particle swarm optimization · Makespan · Energy consumption Right shifting method

M. Nouiri (✉) · A. Jemai
LIP2 Laboratory, Sciences Faculty, University of El Manar of Tunis, 2092 Tunis, Tunisia
e-mail: maroua.nouiri@gmail.com

M. Nouiri
Polytechnic School of Tunis, University of Carthage, 2078 Tunis, Tunisia

A. Bekrar · S. Niar
LAMIH, UMR CNRS 8201, University of Valenciennes and Hainaut-Cambrésis, UVHC, Le Mont Houy, 59313 Valenciennes Cedex, France

A. C. Ammari
MMA Laboratory, INSAT Institute, Carthage University, 1080 Tunis, Tunisia

A. C. Ammari
Renewable Energy Group, Faculty of Engineering, Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_35

1 Introduction

The Flexible Job-Shop Scheduling problem (FJSSP) is a strongly NP-hard problem [1]. It consists in scheduling a set of operations forming jobs on a limited set of machines such that a criteria has to be optimized (makespan, tardiness, etc.). Actually, the performance of schedules released to a job shop may greatly be affected by unexpected disruptions of machines. Taking into account uncertainty will increase the complexity of the FJSSP. Recently, research on production scheduling under uncertainty has attracted substantial attention. Many approaches are proposed to solve such problems. In [2], authors classify the different approaches used to cope with disruptions on proactive, reactive and hybrid methods.

One of the most studied approaches is the predictive-reactive hybrid method which starts by generating a predictive schedule to be executed until disruption occurs [3]. Then, a repair procedure will be launched to handle failures. Rescheduling is one of the repair procedures used in the reactive part of the hybrid approach. It consists of updating an existing schedule in response to disturbances and unplanned events and is a key issue in disruption management [4]. As the problem of rescheduling has a basic importance equal to that of the initial scheduling problem, we propose in this work a rescheduling method that reschedules not only the operations directly affected by a disruption (i.e. assigned to defected resources) but also the operations located on the functional resources. These operations are known as indirectly affected by the disruption related to the precedence constraints with the directly affected operations. The idea is to update an existing production schedule when the state of the manufacturing system makes it infeasible, in such way as to minimize the effect of machine breakdowns for the overall performance and also to increase the schedule stability. We restrict here the possible disruptions to machine breakdowns which refer to the temporary unavailability of a machine. Two assumptions are used: a single machine breakdown and non-resumable mode (i.e. affected operations have to be restarted). The rest of the paper is organized as follows. The literature review of rescheduling approaches addressing job shop under uncertainties is presented in Sect. 2. Sections 3 and 4 define respectively the FJS problem and the PSO meta-heuristic. Section 5 represents the details of the proposed rescheduling method. Experimentation for performance evaluation of the proposed rescheduling approach is reported in Sect. 6. Section 7 describes how the approach can be extended to take into account energy consumption while searching the best reschedule. Finally a conclusion and future work proposals are given with in Sect. 8.

2 Literature Review in Rescheduling

In the rescheduling literature authors propose many techniques and approaches that tackle the problem of uncertainties in scheduling problems [5]. Vieira et al. propose a survey of most applications of rescheduling manufacturing systems [4]. Most of existing rescheduling methods are based on inserting an ideal time equal to the breakdown duration as a basic repair mechanism and then follow the propagation of change to update the start time and the end time of each affected operation. The rescheduling techniques can be classified into three groups: Right-shift rescheduling, total rescheduling and partial rescheduling [4]. The main principle of *Right-shift rescheduling* (RSR) is to accommodate a disruption by globally right-shifting the schedule. If a breakdown disrupts the processing of an operation, then the interrupted operation is right-shifted by the amount of the downtime. All remaining operations are right-shifted by the amount of the downtime [6]. *Total rescheduling* or full rescheduling (FRS) is considered like a scheduling method. However for rescheduling we schedule only a subset of the initial operation set, namely the set of remaining operations. Any job shop scheduling method can be used for this purpose with some minor modification. When considering existing *partial rescheduling* techniques, we can cite the Affected Operation Rescheduling (AOR) which is proposed by Abumaizar and Svestka in [6]. The basic principle behind the concept of AOR is to accommodate any disruption by pushing some operation starting times forward (delaying them) by the minimum amount required. Authors in [7], propose a Modified Affected Operation Rescheduling (mAOR). The developed method uses the same main concept as AOR but considers different types of rescheduling factors other than just machine breakdown (processing time variation, arrival of unexpected job, urgent job). Authors in [8] propose heuristic rescheduling procedures for the job shop that consider tardiness of jobs as main objective and not only makespan and/or stability like the other existing heuristic approaches.

Only few works have addressed rescheduling in the FJSS Problem (FJSSP). Authors in [9] proposed an adaptive representation to select routing policy to face failures. Souier et al. have proposed real time rescheduling metaheuristic to FMS with routing flexibility [10]. They use an Ant colony optimization, genetic algorithms, simulated annealing and tabu search for solving the alternative routing selection problem in real time in order to reduce the congestion in the system.

All these studies cited above used different rescheduling methods according to the type of repair procedure (total or partial), the heuristic method used for the optimization, the objective function minimizing different parameters such as makespan, stability, mean tardiness or multi-objective optimization combining different metrics. The main characteristic of a rescheduling process is to react quickly to unexpected events. However in existing techniques there is no importance granted to the time needed to find the new repaired schedule.

In this work, we propose a partial rescheduling heuristic that minimizes the increase of makespan value after the breakdown of a machine. According to the literature, there is no predictive-reactive hybrid approach that uses the same algorithm to find the original predictive schedule and the new repaired schedule in the reactive part. The first advantage of such an approach is to use the same solution representation on both scheduling and rescheduling problem. In a previous works [1], Nouiri et al. proposed a Particle Swarm Optimization algorithm (PSO) to find a predictive schedule for the FJSSP with minimum makespan. Thus, in this work we propose a rescheduling heuristic while using the same solution code used in PSO algorithm.

3 Problem Formulation

FJSSP is a generalization of Job-Shop Scheduling problem (JSSP), where an operation can be processed on any one of several machines, usually with varying costs, making it closer to the real-world manufacturing environment. The flexibility increases considerably the complexity of the problem as it requires an additional level of decisions to the sequencing one, i.e., machine assignment among the available ones for each operation. We define the flexible job shop scheduling problem formally with the following specifications:

- J set of jobs, J = {*J1*, *J2*, …, *Jn*} is a set of *n* independent jobs to be scheduled. Each job *Ji* consists of a predetermined sequence of operations. O_{ij} is the operation j of job *Ji*.
- M = {*M1*, *M2*, …, *Mm*} is a set of *m* machines. Each machine can process only one operation at a time. Each operation can be processed without interruption during its performance on one of the set of machines. We denote with P_{ijk} the processing time of operation O_{ij} when executed on machine *Mk*. All machines are available at time 0.

In our case, jobs are independent and no priorities are assigned to any job type. Each machine can process only one operation at a time. The predetermined sequence of operations for each job forces each operation to be scheduled after all predecessor operations (precedence/conjunctive constraint). There are no precedence constraints among operations of different jobs and the transportation time is negligible.

4 Particle Swarm Optimization

In the context of machine breakdown while using predictive reactive approach, there are two phases: a *prescheduling phase* (before the breakdown) and a *rescheduling phase* (after the breakdown). Figure 1 illustrates the difference between the predictive and the reactive part of the proposed hybrid approach based PSO algorithm when solving scheduling problem with disruptions.

The PSO algorithm is based on a population of particles; each of the particles represents a candidate solution to a problem and has three main attributes: the position in the search space $x_i(t)$, the current velocity $v_i(t)$ and the best position ever found by the particle during the search Process $x_i^*(t)$. The principle of the algorithm is to move these particles to find optimal solutions. During the search, each particle updates its velocity and position by the following equations [11].

$$v_i(t+1) = w * v_i(t) + c_1 * \left[x_i^*(t) - x_i(t)\right] + c_2 * \left[x_g^*(t) - x_i(t)\right] \quad (1)$$

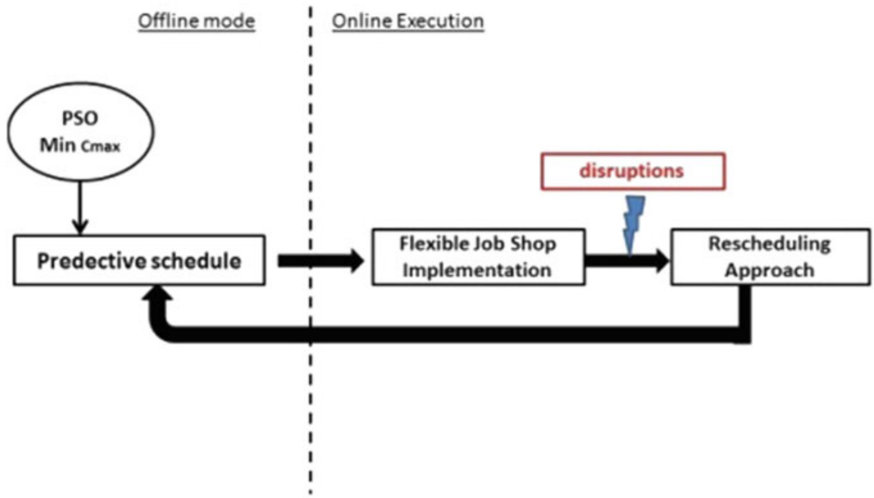


Fig. 1 The predictive reactive approach based PSO algorithm

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

The objective function of the PSO algorithm is to minimize the makespan value of the schedule (Eq. 3).

$$\text{F1} = \text{Min } (\max t_{ij}) \tag{3}$$

where t_{ij} marks the completion time of operation O_{ij}.

We use the PSO algorithm proposed in [1] to generate the predictive schedule. For more details on encoding particle, initialization methods used and tuning parameters one can read [1].

5 New Rescheduling Approach

In this section we present the details of the proposed rescheduling method. We use the routing flexibility to modify the assignment of directly and indirectly affected operations to the others machines. This will be performed either randomly or by choosing the machine that can execute the operation in the minimum operating time. If there is more than one machine that has the lowest operating time, the earliest available is chosen. The rescheduling procedure is as follows:

Step 1: Input parameters: the preschedule $\boldsymbol{p}$, the number of breakdown machine $\boldsymbol{m_b}$, the start time of breakdown $\boldsymbol{st}$, the duration of repair procedure $\boldsymbol{d}$.

Step 2: Extract *subparticle* that contains the directly and indirectly affected operations in order to modify their assignments.

Step 3: Construction of new *subparticles* by reassigning each affected operation on new machines while using the two methods cited above

- Random method (*Alea_iteration*)
- Minimum Earliest method (*1 iteration*)

Step 4: Construction of *swarmReschedule* that contains all particles with the new assignments; reconstruction of particles: the same sequence as the preschedule ***p*** with the new machines assignment of affected operations.

Step 5: Re-evaluate the fitness value F1 (Eq. 3) of all particles of *SwarmReschedule* considering the duration of breakdowns.

Step 6: Select the best particle after reschedule with lowest makespan value.

The first step of the proposed algorithm is to extract *subparticle* that contains both directly and indirectly affected operations to modify their assignments. As the predictive schedule is presented as a particle which is composed of two vectors of process and machine [12], we represent in this work the solution of the rescheduling method also as a particle which is composed of two vectors: Process [ ] defines all affected operations while the Machine [ ] vector represents the machines assigned to this operations. The main differences between the two particles in the scheduling and the rescheduling stage is that the size of two vectors is fixed for the scheduling case and is equal to the total number of operations in the predictive scheduling problem. However, for the rescheduling part the sub particle has a varying size according to the number of directly and indirectly affected operations. This representation enables us to give more flexibility when searching the new solution. A second improvement consists in adding another vector to the encoding procedure which is *MachinesAvailablity*. The size of this vector is equal to the total number of operations and it contains the evolution of time availability of each machine after the processing of each operation. This vector resolves the first affected operation following machine breakdown and also determines the earliest available machine.

In the proposed method we reassign the affected operations using the following two methods:

- **Random Method**: this method reallocates each of the affected operations to a random machine among the available ones with the exception of the failed machine. The variable *Alea_iteration* defines the number of executions of this method to find new *subparticle* with new affectation. Note that if there is only one affected operation in the *subparticle* then the number of not redundant solutions is equal to the total number of machines except the failed one if there is total flexibility, or this number is equal to the number of set machines available for this operation excepted the failed one if there is only partial flexibility.
- **Minimum Earliest Method**: this method generates one solution. For each component of the *subparticle* process, the machine with the lowest processing time for this operation is chosen. If there is more than one machine with

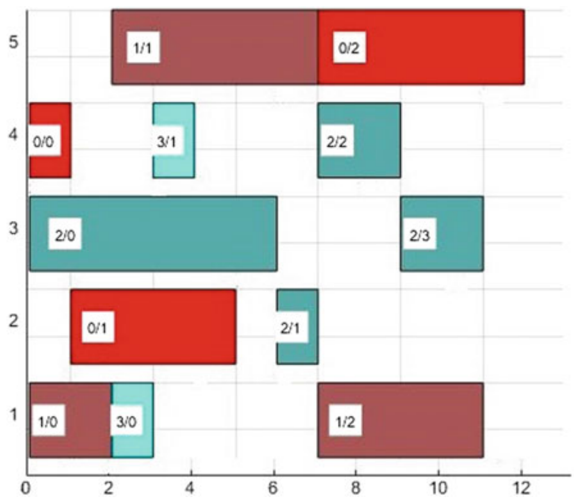


Fig. 2 A predictive schedule obtained by PSO minimises makespan

minimum processing time, then we choose the earliest available one. Note that in the partial FJSSP, if there is one affected operation that can be executed only on the failed machine then systemically we use in this case the RSR rescheduling method [6].

Example: To illustrate the rescheduling heuristic, we consider an example for the flexible shop with 4 jobs and 5 machines. Figure 2 shows the predictive schedule found after applying the PSO algorithm that minimizes the makespan. The make-span value is equal to 12. The particle corresponding to this schedule is illustrated in Fig. 3.

This predicted schedule is subject to the breakdown specified by the yellow rectangle. For example, the historical data indicates that machine M5 has a high risk of failure according to its high busy time and limited reliability.

Figure 4 shows the schedule after applying RSR method. The makespan value is equal to 14.

According to Step 2 in the rescheduling algorithm, we first extract a subparticle that contains the affected operations (see Fig. 5a). The second operation of job 1 and the last operation of job 0 are the directly affected operations by the break-downs because they are located on machine M5 which is in failure. The last operation of job 1 is indirectly affected by the failure according to the precedence constraint. Then the affected operations on new machines are reassigned (Step 3). We obtain Alea_iteration sub particles with new affectation machines after applying random method. Figure 5b illustrates an example of new subparticle obtained by random method. After applying Minimum Earliest method, we obtain only one new subparticle illustrated in Fig. 5c. For the second operation of job 1, there is only one

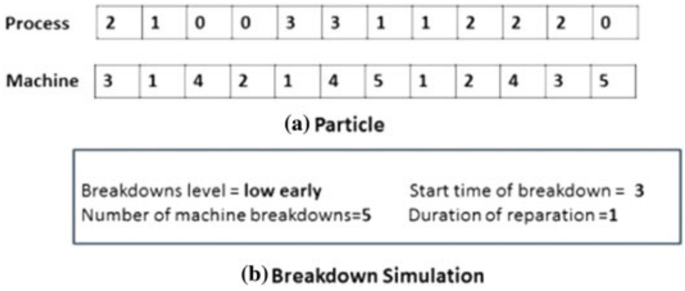


Fig. 3 Particle corresponding to schedule in Fig. 2

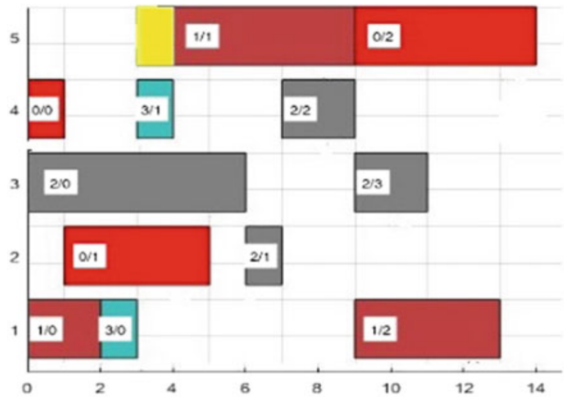


Fig. 4 Schedule after applying RSR method

machine with a minimum processing time M1 while for the last operation of job 1 there are two machines M1 and M3 with the same processing time equal to 4. In this case the earliest available from the MachinesAvailablity vector, machine M3, is chosen (see Fig. 6). Then, based on all obtained subparticles, a swarm of reschedule particles that contains all particles with the different new assignments is constructed (Step 4). In order to reduce the deviation between the original schedule and the repaired one, the sequence of the original schedule is maintained and only the affectation machines of affected operations are changed.

After re-evaluating the fitness value of all particles of *SwarmReschedule* considering the duration of breakdowns, the particle that has the lowest makespan value is chosen. Figure 7 shows the best schedule after applying our rescheduling heuristic. As one can see, the makespan of the repaired schedule is equal to the makespan value of the original schedule 12. However, when we apply the RSR method to the schedule, there is degradation of the makespan value which is equal to 14 (see Fig. 4).

Fig. 5 Subparticle and new subparticles

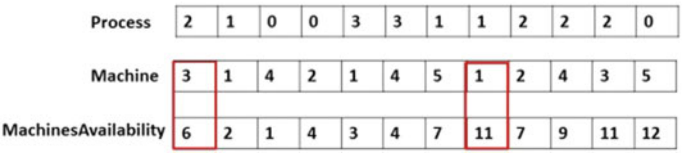
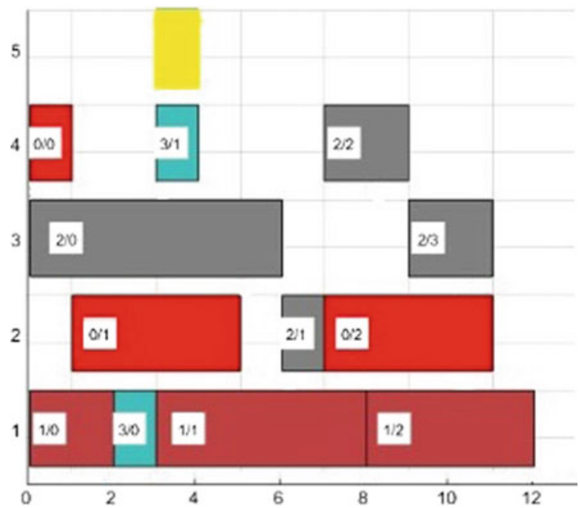


Fig. 6 MachinesAvailablity vector

Fig. 7 Best schedule after applying new rescheduling heuristic



6 Experimentation of the Proposed Rescheduling Heuristic

6.1 Experiment Design

In order to properly experiment the proposed rescheduling method, we need to define exactly how disruption is to be generated and how the performance of the proposed method is evaluated.

6.1.1 Machine Breakdown Formulation

In this work, the machine breakdown generation is simulated in similar manner as in [13]. The probability ρk of a machine M_k to fail is approximated given by the following empirical relation:

$$\rho k = \frac{MBT_K}{MBT_{tot}}$$

where MBT_K is the busy time of machine M_k, and MBT_{tot} is the total busy time of all machines. The breakdown time and duration are generated respectively using the following uniform distributions:

$$\tau_k = [\alpha_1 MBT_K, \alpha_2 MBT_k]$$

$$\tau_{k,duration} = [\beta_1 MBT_K, \beta_2 MBT_k]$$

Where τ_k is the breakdown time of machine M_k, MBT_K is the busy time of machine $M_{k,}$ and the parameters α and β determine the breakdowns type. Table 1 presents the different value of these parameters accordingly to each BD type. The early breakdown happens during the first half of the scheduling horizon and the late one occurs in the second half.

Table 1 Breakdown combinations

Breakdown type	Disruption level	α_1	α_2	β_1	β_2
BD1	Low, early	0	0.5	0.1	0.15
BD2	Low, late	0.5	1	0.1	0.15
BD3	High, early	0	0.5	0.35	0.4
BD4	High, late	0.5	1	0.35	0.4

6.1.2 Rescheduling Performance Measures

In this work we use two types of performance measures proposed in [7]: the *efficiency* and the *stability*.

Efficiency: This measure indicates the effectiveness of the repair in the schedule. It is defined as the percentage change in makespan of the repaired schedule as compared to the original schedule:

$$\eta = \left\{ 1 - \frac{M_{new} - M_0}{M_0} \right\} * 100 \tag{4}$$

where M_{new} is the makespan of the repaired schedule using the new rescheduling method, and M_0 is the makespan of the original schedule. The makespan of the repaired schedule is always greater than or equal to that of the original schedule if the latter does not contain voluntary slacks to absorb disruption. The best repair process is the one in which there is a minimum increase in the makespan after the disruption is incorporated in the repaired schedule. Therefore the maximum efficiency 100% is achieved if the makespan of the original and the repaired schedule are identical.

Stability: The stability of a schedule is measured in terms of deviations of the starting times of job operations from the original schedule. A schedule will be stable if it deviates minimally from the original schedule. The deviation in the starting time is computed as the absolute sum of differences in the starting times of job operations between the initial and repaired schedules. It is then normalized as a ratio of total number of operations in the schedule. A better repair heuristic minimizes the normalized deviation which is defined as:

$$\xi = \frac{\sum_{j=1}^{k} \sum_{i=1}^{p_j} \left| \left(S_{ji}^{*} - S_{ji} \right) \right|}{\sum_{j=1}^{k} p_i} \tag{5}$$

where: ξ is the normalized deviation, p_j is the number of operations of job *j*, K is the number of jobs, S_{ji} is the starting time of the *i*th operation of job *j* in the original schedule, and S_{ji}^{*} is the starting time of the *i*th operation of job *j* in the repaired schedule.

6.2 Experimental Results

To test the effectiveness and performance of the proposed rescheduling algorithm, we carried out experiments with four different benchmark FJSP instances: three with total flexibility (4 * 5, 10 * 10, and 10 * 15) and one with partial flexibility 8 * 8. Each instance can be characterized by the number of jobs (n), number of machines (m), and the related operations Oij associated to each job i.

Due to the limited available text space, we will not present the data of instances. One can see details of instances in [14].

The proposed method is compared to RSR method according to its measure of efficiency and stability. The static schedules generated from PSO algorithm are subject to disruptions of various dimensions. The original schedule is then repaired using the RSR and the proposed rescheduling heuristic. Each instance is tested for four disruption scenarios. The instance is run 10 times on each breakdown scenario. Tables 2, 3, 4 and 5 show the compared results in terms of efficiency and stability targeting BD1, BD2, BD3 and BD4 breakdowns types respectively. These tables consist of 3 columns: the first column represents the instance size. The other columns represent the efficiency and stability of each repaired schedule obtained using our rescheduling algorithm and the RSR method respectively when subject to a specific type of breakdown.

The results indicate that the proposed approach yields significantly better performance than the RSR heuristic in terms of efficiency and stability. A closer look at the different tables reveals that using our proposed method improves the average of stability from 0.601 to 0.394 when facing BD2, from 1.622 to 1.134 when facing BD3, from 1.696 to 0.721 when facing BD4 and increases the average efficiency from 81.354% up to 87.034% when facing BD2, from 77.823% up to 84.54% when facing BD3 and from 63.556% to 91.06% when facing BD4. The increase in the value of the efficiency means that the makespan for the new schedules obtained using RSR was significantly higher than the one obtained using the new rescheduling heuristic. However for the instance 8 * 8 with partial flexibility the RSR technique appears to perform better when facing BD1, BD2 and BD3. When facing BD1, our method gives better results on instances 4 * 5 and 15 * 10.

The proposed rescheduling method gives lower performance for instances with partial flexibility because there is less flexibility for machine assignments in such cases. However when considering BD4, it is shown that the proposed algorithm performs better in all type of instances either with total or partial flexibility. Actually, BD4 breakdown occurs at a late stage and with high duration; thus a

Table 2 Experimental results targeting BD1

Instance	Efficiency (%)		Stability	
	Our method	RSR	Our method	RSR
4 * 5	87.119	86.285	0.5661	0.5666
10 * 10	81.162	90.105	1.709	1.425
15 * 10	93.543	88.647	0.680	0.911
8 * 8	72.67	96.534	0.454	0.291
Average	83.623	90.392	0.852	0.798

Table 3 Experimental results targeting BD2

Instance	Efficiency (%)		Stability	
	Our method	RSR	Our method	RSR
4 * 5	78.9891	69.8762	0.6911	1.507
10 * 10	93.193	75.968	0.029	0.1245
15 * 10	92.074	86.683	0.575	0.679
8 * 8	83.88	92.891	0.281	0.096
Average	87.034	81.354	0.394	0.601

Table 4 Experimental results targeting BD3

Instance	Efficiency (%)		Stability	
	Our method	RSR	Our method	RSR
4 * 5	85.603	68.934	1.482	2.256
10 * 10	92.33	86.996	0.506	0.922
15 * 10	89.339	68.575	1.065	2.36
8 * 8	70.89	86.79	1.483	0.95
Average	84.54	77.823	1.134	1.622

Table 5 Experimental results targeting BD4

Instance	Efficiency (%)		Stability	
	Our method	RSR	Our method	RSR
4 * 5	79.186	45.662	1.336	2.448
10 * 10	98.888	70.216	0.052	1.186
15 * 10	97.705	65.385	1.271	2.494
8 * 08	88.478	72.962	0.225	0.659
Average	91.06	63.556	0.721	1.696

lower number of affected operations on the subparticle is found, which preserves more the stability of the repaired schedule.

In these experiments, the size of the schedule has no major impact on the performance of the rescheduling method; however, the duration of the machine breakdown has a significant effect. If the breakdown duration is small irrespective of occurring early or late in the schedule (BD1, BD2), it is accommodated with relative ease and in this case simple RSR methods can solve the problem with sufficient performance. However, the RSR technique shows poor performance results for breakdowns with high durations. The worst performance result is obtained for BD4 (high-late). In this case, it is very difficult for the RSR to accommodate a bigger disruption at a later half of the schedule and therefore an increase in the makespan is inevitable. Furthermore, one of most important characteristic of the rescheduling repair procedure is the processing time of the method that defines how quickly the repair procedure reacts to the distribution. Thus, the CPU processing time of the proposed rescheduling method is studied for different cases. The obtained results are shown in Table 6.

Table 6 CPU time comparison between proposed rescheduling method and RSR

Instance	Our method CPU times (s)	RSR CPU times (s)
4 * 5	0.255	0.017
10 * 10	0.512	0.02
15 * 10	0.917	0.182
8 * 8	0.453	0.037

As we can see in Table 6, the rescheduling heuristic needs more time to find the most stable and efficient repaired schedule. However, RSR technique has less CPU time to find the solution. This is explainable because the result of RSR is only one reschedule while the proposed method searches many feasible solutions to select the best one. This time varies in practice; the schedule found by RSR method is applied on the shop floor only after repairing the machine in failure. However our heuristic do not require this waiting time for repairing. To reduce the time in the proposed method, the value of Alea_iteration variable must be configurable and adaptive according to the breakdown type and the size of subparticle. In practice, rescheduling is done periodically to plan activities for the next time period based on the state of the system or occasionally. However our method is designed to be event-driven. The rescheduling time of the proposed rescheduling heuristic is similar or smaller than the shortest processing time for all tested instances. Thus it can be neglected and it provides a quickly reaction to failures.

7 Perspectives: Rescheduling to Minimize Energy Consumption

This part discusses some futures works aiming to solve a recent critical objective, which is sustainability. Few studies have used the energy consumption as a criterion to select the best reschedule. Authors in [15] have developed a research framework for energy-efficient scheduling. Tonelli et al. in [16] propose a centralized and distributed model for an off-line energy-aware scheduling problem. In [17], the authors focus on rescheduling in dynamic job-shop scheduling problems where machines can work at different rates. The authors propose a new match-up technique and a mimetic algorithm to find a schedule that minimizes the energy consumption. Many efforts have been made to decentralize control architectures by designing dynamic production scheduling and distributing in an intelligent way the control scheme [18]. Especially there is a focus on using agent-based/holonic-based approaches to develop Intelligent Manufacturing Systems [19]. Authors in [20] proposed an agent-based approach for measuring in real time energy consumption of resources in job-shop manufacturing processes. This work is the first step for developing a new rescheduling heuristic that will be later extended to find a feasible schedule that minimizes both makespan and energy consumption.

8 Conclusion

This work addresses the problem of rescheduling in the FJSSP problem when facing the machine breakdown. A new rescheduling heuristic is proposed. The computational results indicate that proposed method provide superior performances in terms of efficiency and stability than the RSR technique. An interesting direction for future researches is to integrate the developed rescheduling method in distributed framework while using embedded systems in order to find an energy-efficient production rescheduling that is adaptable to new manufacturing challenges. Another interesting direction for future researches is to propose a hyper heuristic based on learning mechanism for selecting and generating the right rescheduling method according to the state of manufacturing system.

Acknowledgements The research work presented in this paper comes from the ELSAT2020 project of CPER sponsored by the French Ministry of Sciences, the Haut de France region and the FEDER.

References

1. Nouiri, M., Bekrar, A., Jemai, A., Niar, S., Ammari, A.C.: An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. J. Intell. Manuf. 1–13 (2015)
2. Chaari, T., Chaabane, S., Aissani, N., Trentesaux, D.: Scheduling under uncertainty : survey and research directions. Int. Conf. Adv. Logist. Trans. 267–272 (2014)
3. Cardin, O., Trentesaux, D., Thomas, A., Castagna, P., Berger, T., El-Haouzi, H.: Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges. J. Intell. Manuf. 1–15 (2015)
4. Vieira, G.E., Herrmann, J.W., Lin, E.: Rescheduling manufacturing systems: a framework of strategies, policies and methods. J. Sched. **6**(1), 39–62 (2003)
5. Katragjini, K., Vallada, E.: Rescheduling flowshops under simultaneous disruptions. Int. Conf. Ind. Eng. Syst. Manag. (IESM). Sevilla, Spain, 21–23 Oct (2015)
6. Abumaizar, R.J., Svestka, J.A.: Rescheduling job shops under random disruptions. Int. J. Prod. Res. **35**(7), 2065–2082 (1997)
7. Subramaniam, V., Raheja, A.S.: mAOR: a heuristic-based reactive repair mechanism for job shop schedules. Int. J. Adv. Manuf. Technol. **22**(9), 669–680 (2003)
8. Dong, Y., Jang, J.: Production rescheduling for machine breakdown at a job shop. Int. J. Prod. Res. **50**(10), 2681–2691 (2012)
9. Unachak, P.: Goodman: adaptive representation for flexible job-shop scheduling and rescheduling. In: Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 511–516. Shanghai, China, 12–14 June (2009)
10. Souier, M., Sari, Z., Hassam, A.: Real-time rescheduling metaheuristic algorithms applied to FMS with routing flexibility. Int. J. Adv. Manuf. Technol. **64**(1), 145–164 (2013)
11. Kennedy, J., Eberhart, R.: Particle swarm optimization. IEEE Int. Conf. Neural Netw. 1942–1948 (1995)
12. Jia, Z., Chen, H., Tang, J.: An improved particle swarm optimization for multi-objective flexible job-shop scheduling problem. Int. Conf. Grey Syst. 1587–1592 (2007)

13. Nouiri, M., Bekrar, A., Jemai, A., Trentesaux, D., Ammari, A.C., Niar, S.: Two stage particle swarm optimization to solve the flexible job shop predictive scheduling problem considering possible machine breakdowns. Comput. Ind. Eng. **112**, 595–606 (2017)
14. Motaghedi-Larijani, A., Sabri-l, K., Heydari, M.: Solving flexible job shop scheduling with multi objective approach. Int. J. Industr. Eng. Prod. Res. **21**(4), 197–209 (2010)
15. Gahm, C., Denz, F., Dirr, M., Tuma, A.: Energy efficient scheduling in manufacturing companies: a review and research framework. Eur. J. Oper. Res. **248**(3), 744–757 (2016)
16. Tonelli, F., Bruzzone, A., Paolucci, M., Carpanzano, E., Nicol, G., Giret, A., Salido, M., Trentesaux, D.: Assessment of mathematical programming and agent based modelling for off-line scheduling: application to energy aware manufacturing. CIRP Ann. Manuf. Technol. **65**(1), 405–408 (2016)
17. Salido, M., Escamilla, J., Barber, F., Giret, A.: Rescheduling in job-shop problems for sustainable manufacturing systems. J. Clean. Prod. 1–12 (2016)
18. Borangiu, T., Răileanu, S., Berger, T., Trentesaux, D.: Switching mode control strategy in manufacturing execution systems. Int. J. Prod. Res. **53**(7), 1950–1963 (2015)
19. Giret, A., Trentesaux, D., Salido, M., Garcia, E., Adam, E.: A holonic multi-agent methodology to design sustainable intelligent manufacturing control systems. J. Clean. Prod. 1–17 (2017)
20. Raileanu, S., Anton, F., Iatan, A., Borangiu, T., Morariu, O.: Resource scheduling based on energy consumption for sustainable manufacturing. J. Intell. Manuf. 1–12 (2015)

On Capacity Measurement in Two Classes of Shop Floor Problems with Multifunctional Parallel Machines

Virginia Ecaterina Oltean, Theodor Borangiu and Silviu Răileanu

Abstract Capacity measurement is of crucial importance in business and manufacturing and is intimately related to both finite and infinite capacity planning. There is a vast literature on this subject and capacity may be defined in various ways. This paper investigates, within two small scale examples, some issues regarding capacity measurement in a shop floor with multifunctional parallel machines that have to process a specified quantity of products of different types, and with specific operations requirements, in a specified working time. In the first example, each operation type has a specific operation unit time, independently on the working machine, while in the second example each machine can execute any operation from its capabilities portfolio in a unique operation's unit time, with a unique associated operation unit cost. The study emphasizes, in the first example, that the capacity measurement depends not only on machines capabilities, on products requirements and on the imposed working time, but also on the allocation strategy of groups of machines to groups of products, while the second example shows that, in case of machines with unique operation unit time for all operation types, the maximal number of operations executable in given working time is a valid capacity measure. The discussed examples may serve as starting point for defining capacity planning procedures for more complex scenarios that can be tested using dedicated software tools, targeting industrial applications.

V. E. Oltean (✉) · T. Borangiu · S. Răileanu
Department of Automation and Applied Informatics, University Politehnica of Bucharest, 060042 Bucharest, Romania
e-mail: ecaterina.oltean@aii.pub.ro

T. Borangiu
e-mail: theodor.borangiu@cimr.pub.ro

S. Răileanu
e-mail: silviu.raileanu@cimr.pub.ro

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_36

1 Introduction

Capacity measurement is an important manufacturing problem occurring in both finite and infinite capacity planning [1], and there is a vast literature regarding capacity management and production scheduling [2–6], to cite only a few of the contributions. The concept of capacity regards the capability of a production unit to produce output in specific time period with available resources [7]; developing adequate tools for capacity measurement is a challenge for both the academic and industrial communities. There is no unique theoretical definition of capacity [1], and, according to [8] one can distinguish between design capacity, as maximum output that can possibly be achieved, effective capacity, as maximum possible output given various constraints, and actual output, as the rate of output actually attained, which is typically less than effective capacity.

This paper discusses, within two theoretical small scale examples, some issues concerning the capacity measurement in shop floors composed of several types of multifunctional parallel machines producing specific quantities of several types of products in a given working time. The first example focuses on building alternative solutions to a heuristic allocation and scheduling problem while the second example treats an exact optimal allocation problem as a preliminary stage used, by a production control department, in future production scheduling scenarios. In the first example each operation type requires a specific execution time unit, and in the second example each multifunctional machine type is characterized by a unique operation time unit, as in case of some multifunctional industrial robots [9]. As in both examples specific constraints are formulated, the capacity measurement regards effective capacity evaluation. The first small-scale example is discussed in Sect. 2, and the second small scale example is described in Sect. 3, followed by concluding remarks.

2 A Heuristic Scheduling Problem in Case of Distinct Operation Times and Multifunctional Parallel Machines

Consider a shop floor containing machines of three types: m_1, m_2 and m_3. The production order demands the execution of two types of products, A and B. Each type of product requires specific operations from a predefined operations set $Op = \{op_j : j = 1:4\}$. Each operation $op_j \in Op$ requires *a specific operation unit time* t_{opj}, *independently* on the machine performing the execution. Also, once an operation execution is started, it cannot be interrupted. For each product, the required operations can be executed in any order but once a product is started to be processed, it has to be finished before launching the execution of another product. The shop floor structure and the corresponding machines capabilities are described in Table 1 and the product types are characterized in Table 2.

Table 1 Machines capabilities with distinct operation times independently on the machine type

Machine types/machines	Operations op_j and their operation unit times t_{op_j} expressed in basic time units [tu], $j = 1{:}4$			
	op_1	op_2	op_3	op_4
$m_1/m_{11}, m_{12}$	8	6	6	10
m_2/m_{21}	8	–	6	–
m_3/m_{31}	–	6	–	10

The machines can be grouped to work in *parallel* but within the group they have to be *coordinated* according to their *capabilities* and to the *required product operations*; this working regime is denoted hereinafter by the operator // applied to the machines in question. It is assumed that any machine working in such a group may interchange a product with other machines in the same group, and consequently idle times may occur, because the operation unit times are not equal (Table 1). The groups of parallel working machines are also considered to work in parallel, but in this case *independently* of other groups, so during their processing the products cannot be interchanged between groups of machines.

The production order demands the execution of four pairs of products of types A and B, denoted $4 \times (A, B)$, without exceeding the working time (or due time) $T_w = 32\,[\text{tu}]$.

The problem is first to identify possible allocation and scheduling variants, and second to choose, among these variants, the ones that can become production execution solutions. The *capacity measurement* is a sub-problem, embedded in the construction and analysis of each of the identified variants.

The heuristic allocation and scheduling obeys also the classic constraints: any machine can process a single product at a time, and any product cannot be processed on two machines simultaneously. The heuristic allocation and scheduling scenarios and alternative solutions are summarized in Tables 3, 4, 5, 6, 7 and 8 and discussed next in brief.

The allocation and scheduling variant 1 proposes to allocate and schedule the execution of $2 \times (A, B)$ products on $m_{11} // m_{12}$ (Table 3) and of $2 \times (A, B)$ products on $m_{21} // m_{31}$ (Table 4). Both groups of parallel machines also work in parallel, but independently of each other.

For both groups of machines, the execution of the allocated products is finished in due time $T_w = 32\,[\text{tu}]$. During the execution process on both machines groups $m_{11} // m_{12}$ and $m_{21} // m_{31}$, non-contiguous idle times occur, caused by the necessity to interchange products to synchronize the machines (Tables 1 and 2).

Table 2 Product types and required operations

Product type	Sets of required operations types	Minimal product execution times
A	$\{op_1,\ op_2\}$	$T_A = t_{op1} + t_{op2} = 14\,[\text{tu}]$
B	$\{op_3,\ op_4\}$	$T_B = t_{op2} + t_{op4} = 16\,[\text{tu}]$

Table 3 Scheduling variant 1: execution of $2 \times (A, B)$ products on machines group $m_{11} // m_{12}$

Product operations on m_{11}	op_3, B_1	op_4, B_1	op_3, B_2	op_4, B_2
Product operations on m_{12}	op_2, A_1	op_1, A_1	op_2, A_2	op_1, A_2
Execution step k	1	2	3	4
Step duration $d(k)$ [tu]	6	$\max\{8,\ 10\} = 10$	6	$\max\{8,\ 10\} = 10$
Cycle duration, idle time	$T^1_{m_{11}//m_{12}} = 2 \cdot (6 + 10) = 32 = T_w$ [tu], $Id^1_{m_{11}//m_{12}} = 2(10 - 8) = 4$ [tu]			

Table 4 Scheduling variant 1: execution of $2 \times (A, B)$ products on machines group $m_{21} // m_{31}$

Product operations on m_{21}	op_3, B_3	op_1, A_3	op_3, B_4	op_1, A_4
Product operations on m_{31}	op_2, A_3	op_4, B_3	op_2, A_4	op_4, B_4
Execution step k	1	2	3	4
Step duration $d(k)$ [tu]	6	$\max\{8,\ 10\} = 10$	6	$\max\{8,\ 10\} = 10$
Cycle duration, idle time	$T^1_{m_{21}//m_{31}} = 2 \cdot (6 + 10) = 32 = T_w$ [tu], $Id^1_{m_{21}//m_{31}} = 2 \cdot (10 - 8) = 4$ [tu]			

Table 5 Scheduling variant 2: execution of $4 \times A$ products on machines group $m_{11} // m_{12}$

Product operations on m_{11}	op_1, A_1	op_2, A_1	op_1, A_3	op_2, A_3
Product operations on m_{12}	op_1, A_2	op_2, A_2	op_1, A_4	op_2, A_4
Execution step k	1	2	3	4
Step duration $d(k)$ [tu]	8	6	8	6
Group cycle duration, idle time	$T^1_{m_{11}//m_{12}} = 2 \cdot (6 + 8) = 28 < T_w$ [tu], $Id^2_{m_{11}//m_{12}} = 0$ [tu]			

The allocation and scheduling variant 2 proposes to allocate the execution of $4 \times A$ products on $m_{11} // m_{12}$ (Table 5) and the execution of $4 \times B$ products on $m_{21} // m_{31}$ (Table 6).

The execution of the $4 \times A$ products on the machines group $m_{11} // m_{12}$ is finished before the due time $T_w = 32$ [tu], but for $m_{21} // m_{31}$ the group cycle duration is longer than T_w, which causes a *failure* of this variant. Also, during the execution process on the group of machines $m_{21} // m_{31}$, a non-contiguous idle time occurs.

As a possible explanation for this failure is the fact that the products of type B, with *longer* minimal execution time T_B than those of type A (Table 2), are allocated to the pair of machines of type m_2 and m_3, respectively, with reduced capabilities compared to the machines of type m_1 (Table 1). In an attempt to correct this effect, a third variant is investigated, in which the two product types are interchanged.

The allocation and scheduling variant 3 proposes to allocate the execution of $4 \times B$ products on $m_{11} // m_{12}$ (Table 7) and the execution of $4 \times A$ products on $m_{21} // m_{31}$ (Table 8).

Table 6 Scheduling variant 2: execution of $4 \times B$ products on machines group $m_{21} // m_{31}$

Product operations on m_{21}	op_3, B_1	op_3, B_2	op_3, B_3	op_3, B_4
Product operations on m_{31}	op_4, B_2	op_4, B_1	op_4, B_3	op_4, B_4
Execution step k	1	2	3	4
Step duration $d(k)$ [tu]	$\max\{6,\ 10\} = 10$	$\max\{6,\ 10\} = 10$	$\max\{6,\ 10\} = 10$	$\max\{6,\ 10\} = 10$
Group cycle duration, idle time	$T^2_{m_{21}//m_{31}} = 4 \cdot 10 = 40 > T_w$ [tu], $Id^2_{m_{21}//m_{31}} = 4 \cdot (10 - 6) = 16$ [tu]			

Table 7 Scheduling variant 3: execution of $4 \times B$ products on machines group $m_{11} // m_{12}$

Product operations on m_{11}	$op_3,\ B_1$	$op_4,\ B_1$	$op_3,\ B_3$	$op_4,\ B_3$
Product operations on m_{12}	$op_3,\ B_2$	$op_4,\ B_2$	op_3,B_4	$op_4,\ B_4$
Execution step k	1	2	3	4
Step duration $d(k)$ [tu]	6	10	6	10
Group cycle duration, idle time	$T^3_{m_{11}//m_{12}} = 2 \cdot (6 + 10) = 32 = T_w$ [tu], $Id^3_{m_{11}//m_{11}} = 0$ [tu]			

Table 8 Scheduling variant 3: execution of $4 \times A$ products on machines group $m_{21} // m_{31}$

Product operations on m_{21}	op_1, A_1	op_1, A_2	op_1, A_3	op_1, A_4
Product operations on m_{31}	op_2, A_2	op_2, A_1	op_2, A_4	op_2, A_3
Execution step k	1	2	3	4
Step duration $d(k)$ [tu]	$\max\{6,\ 8\} = 8$	$\max\{6,\ 8\} = 8$	$\max\{6,\ 8\} = 8$	$\max\{6,\ 8\} = 8$
Group cycle duration, idle time	$T^3_{m_{21}//m_{31}} = 4 \cdot 8 = 32 = T_w$ [tu], $Id^3_{m_{21}//m_{31}} = 4 \cdot (8 - 6) = 8$ [tu]			

For both machines groups working in parallel, the duration of their production cycle is equal to the working time T_w, so the production order $4 \times (A, B)$ can be finished in due time. A 4th variant, in which machines are grouped to work $m_{11} // m_{21}$ and $m_{12} // m_{31}$, that is a machine of type m_1 with machines of type m_2 or m_3, is not a valid option due to a too long duration of the cycle of each group, exceeding T_w.

Summing up, in what concerns the capability measurement problem, one can evaluate only the capacity of groups of machines working in parallel, associated with groups of products of specific types:

- From Tables 3 and 4 it results that the machine groups $m_{11} // m_{12}$ and $m_{21} // m_{31}$ working in parallel have each one a capacity of $2 \times (A, B)$ products in a time $T^1_{m_{11}//m_{12}} = T^1_{m_{21}//m_{31}} = T_w$ and thus an overall capacity of $4 \times (A, B)$ products;
- From Tables 7 and 8 it results that the machine groups $m_{11} // m_{12}$ and $m_{21} // m_{31}$ working in parallel have the capacities of $4 \times B$ and $4 \times A$ products, respectively, in a time $T^3_{m_{11}//m_{12}} = T^3_{m_{21}//m_{31}} = T_w$, with an overall capacity of $4 \times (A, B)$ products;
- In the second variant (Tables 5 and 6), the machines group $m_{21} // m_{31}$ has a too low capacity, as it can process the $4 \times B$ products in a group cycle duration $T^2_{m_{21}//m_{31}} = 40 > T_w$ [tu].

Remark 1 The above analysis was performed assuming that, for each type of products, there is no order imposed to the execution of operations. If product operations must obey an imposed sequencing, then this also influences the capacity measurement because it influences the duration of the production cycles on machines and groups of machines. Acting as an additional constraint, operations sequencing is expected to drive to a diminishing of the capacity reported to the fixed working time T_w.

3 An Exact Allocation Problem in Case of Unique Operation Unit Times for Each of the Multifunctional Parallel Machines

Similarly to the case discussed in Sect. 2, consider a shop floor containing machines of three types, m_i, $i = 1:3$, two of the first type and one of each of the last two types (Table 9). Also, the production order demands the execution of two types of products, A and B, and each type of product requires specific operations from a predefined operations set $Op = \{op_j : j = 1:4\}$, as already specified in the first two columns in Table 2. In contrast to the previous case, in this second case each machine can execute *any* operation from its capabilities portfolio in a *unique operation unit time*, with a *unique associated operation unit cost*. The shop floor structure is given in Table 9.

Table 9 Machines capabilities with unique operation time for each machine type

Machine types/ machines	Executable operations set	Operation unit time t_{mi}[tu]	Operation unit cost c_{mi}[cost unit]
$m_1/m_{11}, m_{12}$	$Op_1 = \{op_j : j = 1:4\} = Op$	$t_{m1} = 6$	$c_{m1} = 0.6$
m_2/m_{21}	$Op_2 = \{op_1, op_3\} \subset Op$	$t_{m2} = 8$	$c_{m2} = 0.8$
m_3/m_{31}	$Op_3 = \{op_2, op_4\} \subset Op$	$t_{m3} = 10$	$c_{m3} = 1$

The production control department investigates what is the minimum cost allocation of machines on product operations in a working time $T_w = 32$ [tu] for a total quantity of $4 \times (A, B)$ products, as a preliminary stage of building several scheduling scenarios reflecting various production and market situations that may occur.

The core of the allocation problem is to express *capacity* and *demand* in the same measurement units, in order to compare them and to build the allocation model. Hence the capacity measurement is an implicit problem that will provide part of the parameters of the allocation problem. This is similar to the approach in [9], and the capacity of each machine type is proposed to be measured *in the number of operations* that can be executed in the working time T_w.

Assumption 1 It will be considered that the two machines m_{11} and m_{12} of type m_1 always work in parallel forming a single production unit or group. The individual machines m_{21}, of type m_2, and m_{31}, of type m_3, are also considered as single machine groups, according to their type, respectively.

Assumption 2 The demand of operations is defined for all the products of same type. The product groups are $4 \times A$ and $4 \times B$.

Remark 2 The philosophy of introducing the previous two assumptions is to group machines and products into classes according to their type, respectively, in order to structure the problem data and to obtain the lowest possible number of decisions variables. In Sect. 2, the groups of machines and of products were built in alternative ways, associated to the proposed scheduling variants. In the case of the optimal allocation problem, the machines and products are grouped in unique ways, according to machines capabilities (Table 9) and to products requirements (the first two columns of Table 2), respectively.

To build the model of the allocation problem, the following parameters are defined:

- The capacity of a machine of type m_i, $i = 1{:}\,3$ is the greatest integer $N_{m_i} = \lfloor T_w / t_{mi} \rfloor$. The capacities values, expressed in number of operations executable in T_w, are $N_{m1} = 10$, for the machines group $m_{11}\ //\ m_{12}$ of type m_1, according to Assumption 1; $N_{m2} = 4$ for m_{21} of type m_2; $N_{m3} = 3$ for m_{31} of type m_3.
- The numbers of demanded operations for the products group $4 \times A$ are: $Nop_{1A} = 4$ operations of type op_1 and $Nop_{2A} = 4$ operations of type op_2 (Table 2).
- The numbers of demanded operations for the products group $4 \times B$ are: $Nop_{3B} = 4$ operations of type op_3 and $Nop_{4B} = 4$ operations of type op_4 (Table 2).

The minimal cost allocation problem is to determine the *integer* numbers x_{ijA}, $j = 1{:}\,2$, and x_{ijB}, $j = 3{:}\,4$, of operations of type $op_j \in Op$ executed by the machines of type m_i, $i = 1{:}\,3$ (Table 9) for the products groups $4 \times A$ and $4 \times B$, respectively, such that: (a) the numbers of demanded operations is satisfied for each product group $4 \times A$ and $4 \times B$, (b) the capacity of each machine group is not exceeded, and (c) the total execution cost is minimized.

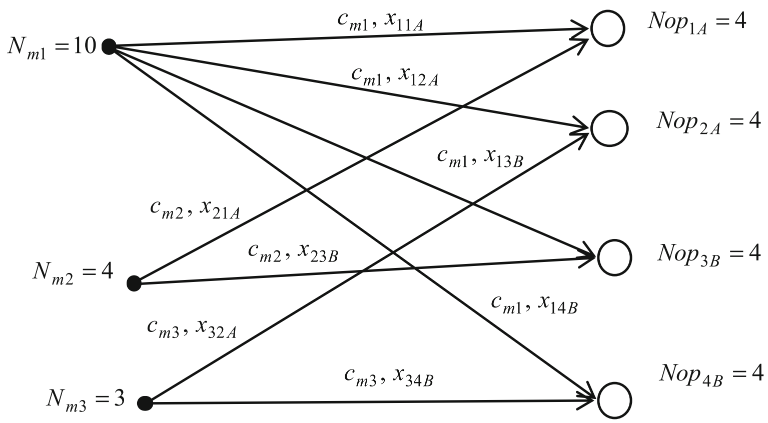


Fig. 1 The digraph of the allocation IP problem (1)

More formally, the integer programming (*IP*) allocation problem is

$$IP: z = \min\{c^T x: Ax \geq b, x \geq 0, x \in \mathbf{Z}^8\} \tag{1}$$

with the associated digraph in Fig. 1 and the vector of decision variables x, the vector of lowest constraints limits b, and the vector of operation unit costs c are given by

$$x = [x_{11A} \quad x_{12A} \quad x_{13B} \quad x_{14B} \quad x_{21A} \quad x_{23B} \quad x_{32A} \quad x_{34B}]^T, \tag{2}$$

$$b = [Nop_{1A} \quad Nop_{2A} \quad Nop_{3B} \quad Nop_{4B} \quad -N_{m1} \quad -N_{m2} \quad -N_{m3}]^T, \tag{3}$$

and

$$c = [c_{m1} \quad c_{m1} \quad c_{m1} \quad c_{m1} \quad c_{m2} \quad c_{m2} \quad c_{m3} \quad c_{m3}]^T, \tag{4}$$

respectively. The matrix of constraints

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix} \tag{5}$$

is totally unimodular and, with the vector b in (3) integral, it results that the associated linear programming (LP) problem

$$LP\colon \min\{c^T x\colon Ax \geq b, x \geq 0, x \in \mathbf{R}^8\} \tag{6}$$

has an integral optimal solution [10], so one can solve the IP problem (1) using LP solving tools. The MATLAB `linprog` routine outputs the following optimal solution to the IP problem (1):

$$x^* = [2 \quad 3 \quad 2 \quad 3 \quad 2 \quad 2 \quad 1 \quad 1]^T \tag{7}$$

and the optimal cost $z = 11.2$ [cost units] is the lowest cost limit for all scheduling programs built starting from these results, since any additional scheduling constraint will drive to costs increase.

Remark 3 The allocation model (1) can be extended in two aspects. Firstly, the groups of machines can be split into individual machines, with consequence on the capacity measurement. Concretely, if the group composed of the two machines of type m_1 with a capacity of $N_{m1} = 10$ operations executable in $T_w = 32\,[\text{tu}]$ is split, then the capacities of the individual machine m_{11} and m_{12} are $N_{m11} = Nm_{12} = 5$ operations. Secondly, the groups of products of types A and B can also be split into individual products, with corresponding value changes of the numbers of demanded product operations. In what concerns the mathematical model of the allocation problem, these extensions drive to an increase of the number of decision variables without change of the optimal cost, in view of the linearity. Model extensions are motivated by the requirements of refined scheduling scenarios based on the allocation solution, but for industrial larger size applications a dramatic increase of the number of decision variables has to be managed.

4 Concluding Remarks

Developing approaches and tools for capacity measurement in production manufacturing is nowadays subject of interest for the academic and industrial communities, and there is a diversity of solutions and studies reported in the literature.

This paper investigates, within two small scale examples, some issues regarding capacity measurement in a shop floor with multifunctional parallel machines that have to process a specified quantity of products of different types, with specific operations requirements, in a specified working time.

In the first example, each operation type has a specific operation unit time, independently on the working machine. For each product, the required operations are allowed to be executed in any order, and once a product is started to be processed, it has to be finished before launching the execution of another product. Three allocation and scheduling variants are heuristically built and analysed, following preliminary constructions of groups of machines that process, in parallel, allocated groups of products. Each machines group can process the allocated products group in a group cycle time interval; the allocated products may be interchanged between the individual machines in the group, according to the required operations and idle times may occur. The machines groups also work in parallel within the shop floor, but independently one of each other. The capacity is measured, for each machines group, by the number of products processed on the machines of the group in due time. The capacity of the shop floor is the overall number of products processed by the composing machine groups in due time. If for any of the machines groups with associated products groups the duration of the group cycle exceeds the imposed working time interval, the capacity of the shop floor is considered to be too low and the allocation and scheduling variant is rejected.

In the second example, each machine can execute any operation from its capabilities portfolio in a unique operation unit time, with a unique associated operation unit cost. The products of different types have to be processed in a given working time with a minimal production cost. To optimally allocate the machines to the product operations, the machines capacities and the products requirements have to be quantified using identical measuring units, in order to compare them. The proposed solution is to define the machine capacity as the maximal number of operations that can be executed in the specified working time, while the product requirements are expressed, for each operation type, as the number of requested operations. The allocation problem is an *IP* linear problem with a typical digraph structure, and the number of decision variables is minimized by defining groups of machines sharing the same capabilities, on one side, and groups of products sharing the same requirements, on the other side.

The study emphasizes, in the first example, that the capacity measurement depends not only on machines capabilities, on products requirements and on the imposed working time, but also on the allocation strategy of groups of machines to groups of products, while the second example shows that, in case of machines with unique operation unit time, the maximal number of operations executable in given working time is a valid capacity measure. Further research concerns the development of theoretical approaches for systematic definition of capacity measurement. Also, the discussed examples may serve as starting point for defining capacity planning procedures for more complex scenarios that can be tested using dedicated tools like IMB-ILOG [11], targeting industrial applications.

References

1. Chen, J.C., Chen, K.-H., Lin, C.-H., Chen, C.-W., Yang, C.-L.: A study of heuristic capacity planning algorithm for weapon production system. Int. J. Electron. Bus. Manag. **9**(1), 46–57 (2011)
2. Schönsleben, P.: Integral logistics management: Planning and control of comprehensive business process. St. Lucie Press, Boca Raton, London (2003)

3. Jiang, J.C., Chen, K.H., Wee, H.M.: A dynamic scheduling model of parallel-machine group for weapon production. Int. J. Adv. Manuf. Technol. **36**(11–12), 1202–1209 (2008)
4. Wortman, J.C., Euwe, M.J., Taal, M., Wiers, V.C.S.: A review of capacity planning techniques within standard software package. Prod. Plann. Control **7**(2), 117–128 (1996)
5. Chen, J.C., Chen, K.H., Wu, J.J., Chen, C.W.: A study of the flexible job shop scheduling problem with parallel machines and reentrant process. Int. J. Adv. Manuf. Technol. **39**(3–4), 344–354 (2008)
6. Bermon, S., Hood, S.J.: Capacity optimization planning system. Interfaces **29**(5), 31–50 (1999)
7. Murray, M.: Measuring capacity in manufacturing. In: The Balance, Small Business—Supply Chain Management (2016). https://www.thebalance.com/small-business-4074055. Accessed 1 July 2017
8. Production capacity law and legal definition. USLegal Dictionary. https://uslegal.com. Accessed 1 July 2017
9. Oltean, V.E., Răileanu, S., Borangiu, Th.: Some aspects concerning a network constrained approach for a class of job-shop process planning. In: Petre, E., Brezovan, M. (eds.) Proceedings of the 20th International Conference on Systems Theory, Control and Computing (ICSTCC), pp. 584–589. Sinaia, Romania, 13–15 Oct 2016
10. Wolsey, L.A.: Integer Programming. Wiley (1998)
11. Optimization modeling with IBM ILOG OPL, Instructor Workbook (2009)

The Method of Calculating the Assembly and Delivery Plan for Groups of Cargoes in the Special VRPTW Problem of Intra-city Food Delivery

Alexander Lada and Petr Skobelev

Abstract A method of calculating the delivery plan concerning the coordination with a storehouse picker's schedule for intra-city food delivery has been analyzed and is being proposed. In order to assemble a client's order storehouse pickers are employed, who, upon notification collect the necessary items for packaging and delivery. The pickers' assembly schedule should be consistent with couriers' intended schedule of travel around the city: delivering packages to clients according to the clients' preferred time windows. It is necessary to calculate a coordinated and conditionally optimal plan for storehouse pickers on the one hand, and for courier drivers on the other, taking into account their mutual interrelationship. At the same time, it is required to be able to provide a desired time window for delivery of the orders to the clients. These calculations result in minimizing the couriers' routes, by taking into account the forecast of traffic congestion at certain times of the day and adaptively redistributing the assembly of packages (order filling) to other stores, if all pickers in the current location are overloaded.

Keywords RP · Adaptive resource scheduling · Transport logistics LTL · Groups of cargoes · Assembly scheduling

1 Introduction

The problem of transport optimization (Vehicle Routing Problem, VRP), which was first described in [1], is one of the most significant tasks of the modern optimization theory. A classification of optimization problems of transport logistics is given in

A. Lada (✉) · P. Skobelev
Samara State Technical University, Samara, Russia
e-mail: ladalexer@gmail.com; lada@kg.ru

P. Skobelev
e-mail: petr.skobelev@gmail.com

A. Lada
SEC Smart Transport Systems, Samara, Russia

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5_37

[2, 3]. A review and classification of VRP problems with the proposed solution is also given in [4].

The transport process in modern transport logistics is divided into two types: FTL and LTL. FTL (Full Truck Load) transportation is characterized by the fact that the carrier has large contracts with customers for reserving a truck as a whole. This means that all the orders are consistently performed by trucks; one truck can simultaneously carry only one order: it loads the first order, drives through the unloading points and unloads all goods, proceeds to the second order, performs it in the same way, goes to the third order, etc. The task of optimizing such transportations can be substantially simplified by taking not into account the possibility of partial loading while constructing the route. The description of FTL transportation models is made in [5]. The solution of the FTL problem was described in [6–8]. On the contrary, LTL (Less than Truck Load) transportation allows parallel execution of orders, i.e. one truck can simultaneously execute several orders: it loads the first order, drives through the its loading points and on the way loads the second order, unloads the first and part of the second, goes to the third order, etc. The task of optimizing LTL transports is much more complicated than FTL, because in such tasks it is additionally necessary to take into account the time windows for the arrival of the truck for loading and unloading; this allows assigning these tasks to the class of VRPTW (Vehicle Routing Problem with Time Windows). Exact solutions to this problem (with the exception of checks) for large volumes of orders and resources are not currently known, despite some replicated methods that provided inaccurate, but of acceptable quality solutions. In general, these are hybrids varieties of branch and bound methods, as, for example the one reported in [9], or some other heuristic methods (genetic algorithms, neural networks, etc.). The authors of this study have experience in solving similar problems [10, 11], and implementing them in multi-agent frameworks [12, 13].

In this paper, the special VRPTW task of calculating a plan for a courier delivery service for food products is considered; in addition, it is necessary to calculate a plan for assembling food into packages according to clients' orders by the employees (pickers) of grocery stores. Adaptability to reschedule the delivery in case of actual deviations from the plan during assembly, due to disturbances, is also required. With this initial specification, the original VRPTW task is further complicated by the need for constant "looking back" at the order-picking plan, which in turn requires a reconfiguration and reconciliation of the delivery plan with the assembly plan. In other words it is necessary to schedule and reschedule the courier route in real time, select the most convenient loading locations (order picking shops), and unload orders to customers along the transport path. It is also necessary to schedule the work of the order pickers, by creating an assembly plan for each order, adjusting it to constantly incoming orders and sending the couriers to pick them timely. Finally, if the courier is stuck in a traffic jam—it can be considered as a mobile warehouse which will be connected to other couriers (e.g., circulating on motorcycles or even bicycles), in order to intercept a late order and deliver the goods on time to the consumer.

In this paper, we consider the problem formulation and the approach to its solution based on mobile resource management. It should be understood that the real business problem is formulated much generally than the mathematical problem above described, because it includes many additional criteria that need to be taken into account in the business process. One of the key requirements for the development is the need for the system to work in real time. In addition, it is necessary to take into account the workload of the order pickers at each moment of time and the forecast of the road situation (traffic jams) at the time of the courier's departure towards the client.

In the first section we consider the formulation of the problem, in the second one we propose an approach to the solution, and in the conclusion section we consider the results of implementation and the perspective of system and application development.

2 The Assembly Problem

In the logistic system of intra-city delivery of food products under the LTL scheme, it is necessary to arrange the reception of orders, their processing, the assembling of goods into packages and their delivery by couriers. To do this, we have to solve two different tasks: create an assembly and delivery plan.

2.1 The Assembly Plan Problem Definition

Suppose we have a number of stores, each based in a particular location. In each store we have a number of assembly workers (pickers). A flow of orders was received from clients. Each order is characterized by the location of delivery and the requested time window and by the number of goods that should be delivered. For each incoming order, it is necessary to determine the place of its assembly in one of the stores, taking into account the current and projected overload of pickers. We consider that any good is available in any store with infinite quantity.

The goal is to schedule the order assembly in the closest delivery location store taking into account the real time speed of assembly, and change the store to the next closest one if, according to its schedule in the store, it is impossible to deliver it in the requested time window. The task is solved by calculating the planned start and finish times of the assembly according to the method presented below.

2.2 The Assembly Plan Construction Method

It is assumed that the actual start and end time of the order are sent to the scheduling system directly by the external information system in real time (e.g. from the mobile application for pickers). In addition to the start and finish times, the system calculates the planned start and finish times for all orders based on the assembly speed of the item in the store (this depends on the number and productivity of the pickers). It is assumed that each time getting information about a new event related to assembling the goods in a particular store, allows recalculating the current speed of assembly and the planned times of starting and finishing the assemblies not yet collected.

It is assumed that the assembly times of all orders in the store are calculated in parallel queues. Their scheduling is determined according to the arrival time of each order in the system, and corrected by the actual times of the beginning and end of the assembly. The number of these queues is equal to the number of pickers in the store. Within each queue, orders are executed sequentially.

An example is further considered: suppose we have a store with 2 pickers and a table of equivalent orders. Initially we will consider that the assembly time of each order performed by one picker is 5 min (Table 1).

Orders O1 and O2 are collected in parallel, since there are two pickers in the store that have the same estimation on the time of the beginning and end of the assembly. Order O3 should wait until either the first or the second picker is available and therefore will be collected only after the first two orders are completed. Order O4 was received at 13:15; by this time all the pickers have already been released and only one of them will execute the order, because it is imposed that two pickers do not collect the order at the same time.

Next, consider the example of Table 2, showing how the scheduled start/end times of an assembly will be corrected when the actual data is received.

At the time 13:10 the start of assembly was received from O1, which started 10 min later than planned and consequently will be collected now at 13:15; this

Table 1 Chronological order processing

Order no.	Order time	Scheduled time to start the assembly	Scheduled time to finish the assembly	Actual time the assembly started	Actual time the assembly finished
O1	13:00	13:00	13:05		
O2	13:00	13:00	13:05		
O3	13:00	13:05	13:10		
O4	13:15	13:15	13:20		

Table 2 Chronological table for processing orders after the actual start of order assembly O1

Order no.	Order time	Scheduled time to start the assembly	Scheduled time to finish the assembly	Actual time the assembly started	Actual time the assembly finished
O1	13:00	**13:10**	**13:15**	13:10	
O2	13:00	**13:10**	**13:15**		
O3	13:00	**13:15**	**13:20**		
O4	13:15	13:15	13:20		

Table 3 Chronology for processing orders after the actual start of order assembly O1

Order no.	Order time	Scheduled time to start the assembly	Scheduled time to finish the assembly	Actual time the assembly started	Actual time the assembly finished
O1	13:00	13:10	**13:20**	13:10	13:20
O2	13:00	**13:20**	**13:30**		
O3	13:00	**13:20**	**13:30**		
O4	13:15	13:30	13:40		

Table 4 Chronology for processing orders after the actual start of order assembly O2

Order no.	Order time	Scheduled time to start the assembly	Scheduled time to finish the assembly	Actual time the assembly started	Actual time the assembly finished
O1	13:00	13:10	13:20	13:10	13:20
O2	13:00	**13:25**	**13:35**	13:25	
O3	13:00	**13:25**	**13:35**		
O4	13:15	**13:35**	**13:45**		

leads to changes in the planned start and end times for the remaining orders. According to the plan, order O2 will be collected only at 13:10. Order O3 is shifted with 5 min. According to the plan of order O4, no changes will occur, because it has some extra time in reserve.

New factual data was received (Table 3).

At the time 13:20 the fact of assembly completion was received from O1, which was completed 5 min later than planned, and, therefore, the picker's assembling time was changed (instead of the initial 5 min now 10 min), which led to changes in the planned starting points and finishing time of the assembly of other orders.

Order O2 will be collected according to the plan only at 13:20, because it's already 13:20, and the event signaling the beginning of the assembly is not yet received; taking into account the new speed, the assembly will take 10 min, so it will end at 1:30 pm. Order O3 is collected in parallel with the order O2. Order O4 in the plan will move and will begin to be collected immediately after the order O2 or O3.

New factual data was received (Table 4).

At the time 13:25 the event signaling the beginning of the assembly of the order O3 (for convenience, orders with facts immediately go up the queue) is received; in fact the start of the assembly was 5 min later than planned, which led to changes in the planned start and end times for the remaining orders.

New factual data was received (Table 5).

At the time 13:30 the event signaling the beginning of the assembly of O2 order and the completion of the assembly about ordering O3 is received; now the order assembly rate is estimated as the average value between the difference resulting from the two events (the beginning and the end already received), and will be ((13:20−13:10) + (13:30−13:25))/2 = 7.5. The calculations are then carried out in a similar way.

Table 5 Chronology for processing orders after the actual completion of order assembly O2

Order no.	Order time	Scheduled time to start the assembly	Scheduled time to finish the assembly	Actual time the assembly started	Actual time the assembly finished
O1	13:00	13:10	13:20	13:10	13:20
O2	13:00	13:25	**13:30**	13:25	13:30
O3	13:00	**13:30**	**13:38**	13:30	
O4	13:15	**13:30**	**13:38**		

2.3 Choosing the Best Store to Assemble the Order

After calculating the planned finish time for the order assembly for which the store is currently determined, we check for each store whether the courier will reach the client theoretically in the interval from that time to the end of the unloading window (taking into account the time of arrival from the current store to the client). If it does, the result for this store is estimated as possible and saved. From all possible options of stores, we should choose the one that requires the minimum trip time from it to the client (that is, the minimum of the driver's work in the form of mileage), provided that the planned assembly time in this store does not exceed the loading window; otherwise, we consider a more remote store, etc., till the end of the list of shops. We can reason similarly after we make the choice if the picker did not finish the assembly of the order before the deadline (skip the deadline for the assembly), and try to plan the assembly of this order in another store.

3 The Delivery Problem

After completing the selection of the store, we create the delivery schedule for each courier taking into account the number of available couriers, the orders' delivery time window, and the trip time along the route in the city with traffic jams forecasting. We also should adaptively change the plan according to real time events (delays during orders delivery process).

3.1 The Delivery Plan Problem Definition

Suppose we have a number of couriers, each of them being characterized by maximum carried volume and payload. The location for each courier can be evaluated at any moment of time. We have a set of LTL-orders each of them being characterized by the package volume and weight and having only one location of loading (the store location) and one location of unloading (the client location). All orders must be delivered no later than an imposed window time; delays are not considered acceptable. We also consider as acceptable finishing the delivery work exactly at the termination of the window (i.e., the start and end boundaries of the time windows are included). The trip time based on distances between all locations with traffic jams forecasting can be evaluated at any moment of time. There are external data providers for traffic forecast congestions. The most popular providers in Russia are Yandex [14] and Google [15].

The scheduling system requests the time for the route from point A to point B from the "jam" service for the time of the planned delivery; having received the result, the system uses it in future planning as the time of the execution of the route in the process of fulfilling the order. The goal is to allocate orders for available couriers with a minimum total execution time, while maximizing the number of delivered orders and possibly reducing the total trip distance.

3.2 The Delivery Plan Construction Method

For the construction of the plan, the following method is proposed. All orders are sorted by the loading time (the estimated time when the assembly ends). At the initial time, we suppose that a particular courier is allocated in the first order loading point (considered shop). Let the courier in this point appear at the start time T_0, in advance of the earliest loading time of the first order. With respect to this starting point and the initial time T_0, the table of the possible assignments is calculated. In the table, each potential order execution time is calculated according to the formula OET_i (order execution time) = TL_i (trip to loading) + WL_i (work to load) + TU_i (trip to unloading) + WU_i (work to unload). If OET is within the order unloading time window, the order is considered as possible for delivery.

In the generated table we select the order with the shortest OET. After that we change the location of the considered courier, which now corresponds to the chosen place of loading order, and calculate the time of its release as $T_1 = TL_i + WL_i$.

After the first assignment, we generate the table again, but now we exclude some orders which match one of the following situations:

1. The summed volume and weight for previously assigned orders with the considered one added exceed the courier's maximum transportable volume and weight.
2. There is no time for the courier to move from the current location to the considered loading one because it is incompatible with the considered order loading time window.
3. With regard to the considered order pick up, the courier has no time to unload previously assigned orders.

After filtering orders, for the remaining ones calculate again their execution times, considering also the waiting times before opening loading windows with new locations of the courier; we choose the shortest execution time again. Then the algorithm is repeated until there are no more orders that can be possibly allocates because of above described restrictions. Then, the next considered courier is placed in the first order loading point, and the operation is repeated for the remaining orders until there is no order left for allocation.

3.3 Changing the Plan According to Real Time Events

There are two types of possible problems during the order delivery by courier: *delay* and *unavailability* (crash, accident, etc.). A delay problem may occur when the courier moves with orders on the route (e.g., entering into a traffic jam or facing a car defect). A delay may also occur when the courier arrives at the place and begins unloading the order, but some difficulties arise in this process (e.g., closed territory, necessity of a pass, not working elevator, necessity to climb the stairs, etc.). In this case, orders already carried by the courier will be potentially delivered with delay but the next orders that should be loaded by this courier in the future will be rescheduled to other couriers according to the same method.

Unavailability is a more important problem, which means that the courier can no longer perform the delivery (because of crash, accident, etc.). In this case, the courier requests help by creating an additional loading point in the place where he stopped moving and from where he cannot continue to carry out his orders. As a result, a new loading point with the orders of this courier is created. These orders will be rescheduled to other couriers according to the same method.

4 Experimental Results

We performed several experiments to test the described method by using the real data from our client company [16]. The goal of the experiments was to compare the proposed construction delivery plan method with the brute force method. It was

Table 6 Chronological table of orders with details in unified units

	LWs	LWf	LT	ULWs	ULWf	ULT	CL
O_1	1	2	1	6	9	1	5
O_2	2	3	0	6	9	0	4
O_3	3	5	1	12	14	1	5
O_4	3	5	1	12	14	1	4
O_5	4	6	0	17	19	0	4

where:
- LWs—start loading window
- LWf—end loading window
- LT—loading working time
- ULWs—start unloading window
- ULWf—end unloading window
- ULT—unloading working time
- CL—cargo load capacity

Table 7 Matrix of distances between orders loading and unloading points in unified units [u.u]

	LO_1	LO_2	LO_3	LO_4	LO_5		LO_1	LO_2	LO_3	LO_4	LO_5		ULO_1	ULO_2	ULO_3	ULO_4	ULO_5
LO_1		0.2	0.3	0.4	0.5	ULO_1	3	2.5	3	2.5	2.6	ULO_1		0.3	0.4	0.5	0.4
LO_2	0.2		0.6	0.5	0.5	ULO_2	5	4	6	4	4	ULO_2	0.3		0.2	0.3	0.4
LO_3	0.3	0.6		0.4	0.3	ULO_3	4	5	6	4	3	ULO_3	0.4	0.2		0.1	0.5
LO_4	0.4	0.5	0.4		0.1	ULO_4	6	7	5	6	4	ULO_4	0.5	0.3	0.1		0.6
LO_5	0.5	0.5	0.3	0.1		ULO_5	7	5	6	4	3	ULO_5	0.4	0.4	0.5	0.6	

where: LO_i—order loading point, ULO_i—order unloading point

Table 8 Matrix of distances between couriers initial points and orders loading points in [u.u]

	LO_1	LO_2	LO_3	LO_4	LO_5
C_1	0.3	0.4	0.5	0.4	0.6
C_2	0.5	0.2	0.3	0.4	0.7

where: C_i—starting point of courier with a load capacity of 25 units (all the orders can be potentially delivered by one courier because the total sum is CL = 5 + 4 + 5 + 4 + 4 < 25)

Table 9 The experimental results table

Number of orders	Number of couriers	Proposed method result in [u.u]	Proposed method calculation time in seconds	Brute force method result in [u.u]	Brute force method calculation time in seconds
5	2	17	0.001	17	10
10	2	25	0.001	24	1983
15	3	30	0.001	28	19865
20	4	50	0.001	49	167541
25	5	70	0.001		
30	5	85	0.001		
35	6	95	0.001		
40	7	111	0.001		
45	8	122	0.001		
50	9	131	0.003		
55	9	139	0.001		
60	10	153	0.001		
65	10	165	0.003		
70	12	195	0.001		
75	15	205	0.011		
80	16	223	0.014		
85	17	238	0.01		
90	17	243	0.01		
95	20	260	0.08		
100	20	284	0.08		

possible to obtain the brute force results only for the first four experiments at a suitable time. The data related to the experiments is given in Tables 6, 7, 8 and 9.

5 Conclusion

The method and solution for the assembling and delivery problem for groups of cargoes in the special VRPTW problem of intra-city food delivery are developed and described in detail. Based on the developed method, intelligent dispatching management software was created which is used in the daily work of the food delivery service of the Instamart company [16] (the authors received a letter of gratitude). This allowed reducing the average assembly time of the order by 15% mainly by taking into account the actual assembly speed of each store picker and adaptation to the current situation by redirecting the order assembly to another store. From the delivery part this allowed to reduce the average number of delays by 22% because of quick delivery route evaluation for couriers, considering traffic forecast situation and reacting to emergency situations during delivery. When one courier faces a problem, it is possible to solve it by help of other couriers by adaptively changing their current delivery plan. The proposed approach complements a collection of methods and tools developed in [17, 18].

Acknowledgements This work was supported by Russian Foundation for Basic Research, project number 16-01-00759.

References

1. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Manag. Sci. **6**(1), 80–91 (1959)
2. Browne, M., McKinnon, A., Cullinane, S.: Green Logistics. In: Anthony Whiteing (ed.). Kogan Page (2010)
3. Bertazzi, L., Savelsbergh, M., Speranza, M.G.: Inventory routing. In: Golden, B., Raghavan, S., Wasil, E. (eds.) The Vehicle Routing Problem: Latest Advances and New Challenges, vol. 43, pp. 49–72. Springer Science + Business Media, New York (2008). https://doi.org/10.1007/978-0-387-77778-8
4. De Jaegere, N., Defraeye, M., Van Nieu, V.: The Vehicle Routing Problem: State of the Art Classification and Review. KU Leuven (2015). https://lirias.kuleuven.be/bitstream/123456789/457452/1/KBI_1415.pdf
5. Granichin, O., Skobelev, P., Lada, A., Mayorov, I., Tsarev, A.: Cargo transportation models analysis using multi-agent adaptive real-time truck scheduling system. In: Proceedings of 5th International Conference on Agents and Artificial Intelligence (ICAART'2013), vol. 2, pp. 244–249, 15–18 Feb 2013. SciTePress, Barcelona, Portugal
6. Skobelev, P., Lada, A., Mayorov, I.: Finding an initial plan of transport resources FTL allocation in a special VRP problem using linear programming methods. In: Proceedings of the 19th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI), Orlando, Florida, vol. 1, pp. 16–21, 12–15 July 2015. ISBN 978-1-941763-24-7
7. Skobelev, P., Lada, A.: A solution to the subtask of initial distribution of transport resources in a special optimization FTL transportation problem in real-time using the Hungarian algorithm. Indian J. Sci. Technol. **9**(12), 1–8 (2016). https://doi.org/10.17485/ijst/2016/v9i12/89520
8. Lada, A., Sazonov, V.V., Skobelev, P.: Method for transportation cost calculation on the basis of full cycle (round trip). Indian J. Sci. Technol. **9**(20), 1–8 (2016). https://doi.org/10.17485/ijst/2016/v9i20/94478
9. Bortfeldt, A.: A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints, Andreas Bortfeldt—Diskussionsbeiträge der Fakultät für Wirtschaftswissenschaft der Fern Universität in Hagen (2010)
10. Sazonov, V.V., Skobelev, P., Lada, A., Mayorov, I.: Application of multi-agent technologies in the transport task with time windows and several loading points, Management of large systems, issue 64, pp. 65–80. IPP RAS, Moscow (2016)

11. Sazonov, V.V., Skobelev, P., Lada, A., Mayorov, I. Multi-agent approach to the solution of the transport problem with several loading points and time windows. In: Proceedings of the XVIII International Conference “Problems of Control and Modeling in Complex Systems”, Samara, pp. 171–179, 20–25 Sept 2016
12. Skobelev, P., Lada, A., Tsarev, A.V.: Development of intelligent management system for groupage cargoes transportation in real time, Bulletin of the Samara State Technical University, series “Technical Sciences”, No. 3(39), pp. 65–74 (2013)
13. Skobelev, P., Lada, A.: SMART LOGISTICS: multi-agent system for groupage cargoes managing in the intraregional deliveries. In: Proceedings of the XVI International Conference “Problems of Control and Modeling in Complex Systems”, Samara, pp. 253–261, June 30–July 03 2014, SSC RAS. ISBN 978-5-93424-703-5
14. https://tech.yandex.ru/maps/doc/jsapi/1.x/mod/concepts/traffic-docpage. Accessed 12 Apr 2017, Traffic jams-Yandex Technologies
15. https://enterprise.google.ru/intl/ru/maps/products/mapsapi.html. Accessed 12 Apr 2017, Maps API: creation of applications and analytics
16. Home delivery products in the online store Instamart. Buy products with free delivery in Moscow. https://instamart.ru/. Accessed 12 Apr 2017
17. Vittich, V.A., Skobelev, P.: Multi-agent interaction models for building needs networking and opportunities in open systems, pp. 177–185. Automation and Remote Control, Moscow (2003)
18. Rzevski, G., Skobelev, P.: Managing Complexity. WIT Press, UK (2015). http://www.amazon.com/Managing-Complexity-G-Rzevski/dp/1845649362/ref=sr_1_3?s=books&ie=UTF8&qid=1396421715&sr=1-3&keywords=managing+complexity

Author Index

A
Ammari, Ahmed Chiheb, 461
André, Pascal, 359
Anton, Florin Daniel, 201
Anton, Silvia, 201
Aubry, Alexis, 345

B
Babiceanu, Radu F., 243
Barbosa, José, 13, 157
Bekrar, Abdelghani, 317, 395, 461
Berger, Thierry, 187, 291
Bertani, Filippo, 171
Bonte, Thérèse, 291
Borangiu, Theodor, 201, 253, 477
Branger, Guillaume, 93, 111
Bril El Haouzi, Hind, 433
Bril, Hind, 395
Brintrup, Alexandra, 51
Burlat, Patrick, 447

C
Cardin, Olivier, 267, 359, 407
Chargui, Tarik, 317
Coelho, João P., 13
Coullon, Hélène, 217
Cruz Salazar, Luis Alberto, 139

D
Defoort, Michael, 395
Demartini, Melissa, 171
Demesure, Guillaume, 381, 395
Derigent, William, 79, 127
Djemaï, Mohamed, 395
Duflot-Kremer, Marie, 345

E
El Haouzi, Hind Bril, 381

G
Gaudreault, Jonathan, 433
Geraldes, Carla A.S., 13
Ghomri, Latéfa, 407
Gorecki, Simon, 281

H
Haouzi, Hind Bril El, 63
Herakovic, Niko, 371
Himmiche, Sara, 345
Huet, Xavier, 421

I
Indriago, Carlos, 407

J
Jaegler, Anicia, 447
Jaegler, Yann, 447
Jemai, Abderrazak, 461

L
Labarthe, Olivier, 27
Lada, Alexander, 489
Lamouri, Samir, 447
L'Anton, Anne, 267
Lehn, Rémi, 421
Leitão, Paulo, 13, 157
Leray, Christophe, 421
Li, Hang, 139

M
Marangé, Pascale, 345
McFarlane, Duncan, 51
Mebarki, Nasser, 331
Mekki, Kais, 79
Mezgebe, Tsegay Tesfay, 381
Montoya-Torres, Jairo R., 41
Montreuil, Benoit, 27

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence 762,
https://doi.org/10.1007/978-3-319-73751-5

Morariu, Cristina, 253
Morariu, Octavian, 253
Morin, Michael, 433
Muñoz-Villamizar, Andrés, 41

N
Niar, Smail, 461
Nouiri, Maroua, 461
Noyé, Jacques, 217
Noyel, Mélanie, 63

O
Oltean, Virginia Ecaterina, 477
Ormazábal, Marta, 41
Ounnar, Fouzia, 27, 303

P
Panescu, Doru, 229
Pannequin, Rémi, 63, 381
Parrein, Benoît, 421
Pascal, Carlos, 229
Pedrotti, Benoit, 421
Perry, Nicolas, 281
Pétin, Jean-François, 345
Piel, Antoine, 421
Puchkova, Alena, 51
Pujo, Patrick, 27, 303

R
Răileanu, Silviu, 201, 253, 477
Rault, Raphaël, 1
Reghioui, Mohamed, 317
Rondeau, Eric, 79

S
Sallez, Yves, 187, 291
Santos, Javier, 41
Seker, Remzi, 243
Selma, Cyrine, 433
Skobelev, Petr, 489
Srinivasan, Rengarajan, 51

T
Teixeira, Joy, 157
Thomas, André, 63, 79, 127, 381, 395
Thomas, Philippe, 63, 433
Thorne, Alan, 51
Tonelli, Flavio, 171
Trentesaux, Damien, 1, 93, 111, 317, 395, 447

V
Vo, Nhat-Vinh, 291

Y
Yahouni, Zakaria, 331

Z
Zacharewicz, Gregory, 281
Zerovnik, Janez, 371
Zimmermann, Emmanuel, 63
Zupan, Hugo, 371