Pierre-Jean Charrel
Daniel Galarreta
Editors

Project Management and Risk Management in Complex Projects

Studies in Organizational Semiotics

Springer

PROJECT MANAGEMENT AND RISK MANAGEMENT
IN COMPLEX PROJECTS

Project Management and Risk Management in Complex Projects

Studies in Organizational Semiotics

Edited by

PIERRE-JEAN CHARREL

Université Toulouse 2
Institut de Recherche en Informatique de Toulouse, France

and

DANIEL GALARRETA

Centre National d'Etudes Spatiales, Toulouse, France

Springer

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN-13 978-1-4020-5836-3 (HB)
ISBN-13 978-1-4020-5837-0 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springer.com

Printed on acid-free paper

Contents

Preface

The Editors: Pierre-Jean Charrel, Daniel Galarreta – October 2006

1 Project Management and Risk Management in Complex Projects – Studies in Organizational Semiotics

The study of space gives rise to very complex projects calling for contributions from many varied communities of knowledge and practical expertise. These different cultures and their specialized languages define separate *information fields* that generate problems for communication and collaboration, even during the development of technical objects, but especially in the phases of project definition, systems requirements engineering, and design.

These technical objects are not given, *a priori*: they have never been realized before. On the contrary, they are progressively built up by negotiating the meanings of terminologies, formulas, drawings and other representations of artefacts intended to satisfy the many agreed requirements, mechanical, electrical thermal etc. In other words – at least before their construction – these objects have no concrete existence but are semiotic objects; and even when built, their projected *behaviour* in distant corners of space will be known to us only as semiotic constructs.

Organizational semiotics (OS) offers a framework for understanding the processes that this project work entails, in particular the interaction between individuals, between groups, within society, as well as between human and technology. "One of the aims of Organizational Semiotics is showing what you are doing when you are trying to understand, design or change organizations in terms of the use of for instance models and metaphors" [1].

Holding the 8th session of the annual Organizational Semiotics (OS) Workshop in Toulouse – the French capital of aeronautics and space – presents us with the opportunity to test ideas from OS against the problems generated by very complex projects on the frontiers of engineering where complexity and risk are major factors. CNES (Centre National d'Etudes Spatiales), through its Technical Competency Centre in Management (CCT MAN), has provided two excellent, illustrative cases in space exploration (cf. section 2).

To stimulate the contribution of OS ideas, CNES has emphasized two important issues:

- The management of complex highly innovative, multidisciplinary projects during their early volatile phases
- The management of risks faced by such projects that may run far into the future and beyond human intervention

Twelve articles written by 28 authors belonging to several scientific fields are the result of the present call for contributions. As usual, all papers were discussed after their presentation and revised for the sake of the present book.

2 The Case Studies

2.1 Case Study 1: IASI Project

Project management in space activity: an introduction

A project is to go from designing to building.

In order to reach this goal, we need competencies and means to accomplish the different steps of the process, included the ability to co-ordinate these steps. Information (in the different forms of data/information/knowledge) and their processing are the principal elements of this process. The particular attention to the way the project uses its informational resources in order to achieve a success, could be a definition of Project Management activity.

A more classical definition of Project Management is the controlling of the evolution of all the aspects of the project, including Time, Resources and Risks. But this definition implies that semiotic/informational devices (such as relevant indicators, plans...) are available in order to:

- anticipate planned events
- permanently adjust means and constraints
- start actions to preserve sufficient margins
- communicate (inside and outside the project) to manage conflicts, and motivate the teams

In the case of large projects, such as space projects, the description of the information system as a whole is difficult because we are faced the heterogeneity of the organization. Prime contractors, manufacturers, and customers constitute different aspects of this organization. It is not therefore easy to guarantee its efficiency: it is the purpose of the project management activity.

The IASI project

IASI (Infrared Atmospheric Sounding Interferometer) is a significant technological and scientific step forward that will provide meteorologists with atmospheric emission spectra to derive temperature and humidity profiles with a vertical resolution of 1 kilometre and accuracies of 1 Kelvin and 10% respectively.

The first flight model is scheduled for launch in 2006 onboard the METOP series of European meteorological polar-orbiting satellites. CNES is leading the IASI program in association with EUMETSAT (Europe's Meteorological Satellite Organization). CNES has technical oversight responsibility for the instruments up to the end of in-orbit commissioning. It will develop the Data Processing Software which will be implemented in the EUMETSAT Polar System ground segment and will develop and operate a Technical Expertise Centre. EUMETSAT is responsible for operating the instrument and the associated data processing, archiving and distribution to users.

In 1998, CNES and Eumetsat awarded Alcatel Space with the development and production of three IASI instruments which will be carried on the Metop satellites. (All the information about the IASI project can be found on the CNES site: www.cnes.fr in the *CNES programmes* entry, then *Sustainable development* entry*, IASI* sub-entry)

The cooperation between CNES and EUMETSAT started in 1997 and the final version of the Cooperation Agreement was signed in 2001. In the reached agreement CNES is responsible for developing and providing three flight models, data processing software and the technical expertise centre whereas EUMETSAT is in charge of operational exploitation of IASI. This agreement includes conditions on costs sharing, payments, and prices revisions. On technical level the agreement defines the tasks, responsibilities of the parties, the management plan, deliveries, and planning.

The management of the project is based upon the Management Plan document which includes: documentation management, delay management, actions management, description of the supplies, process of the reviews, configuration control, and product breakdown structure.

Then for the daily management, different management charts are used such as the planning, the financial budget, the instrument performance budget, the critical elements list. All these charts are living along the project development.

It is important in such a project to define the responsibilities of each one: the Project Organization Note defines for each activity one leader who is clearly identified and acknowledged by all, and gather all the transverse roles within the hierarchical project structure. It is *very important* to create a project culture.

It is also very important in such a project to avoid designing a solution just for the sake of technology: a link should be permanently maintained with the users in order to develop an instrument which will deliver attractive data for meteorological and scientific communities.

One of the most significant management issues of any project developed in cooperation is to overcome the inertia in the decision process when several entities are involved in development (prime contractor and several agencies). In some case anticipation of the decision was necessary for saving delays necessary but in any case transparency and confidence has always been achieved.

Considering that CNES is concerned by the development of IASI and EUMETSAT is concerned by the exploitation, the relation between CNES and EUMETSAT could be seen as supplier/customer relationship. In fact, due to a good confidence established between CNES and EUMETSAT, a partnership relation prevails over a supplier/customer relationship.

2.2 Case 2: Risk Management and the Rosetta Project

Risk management in space activity: an introduction

The complex character of the organization of large projects gives a new vision about the risk notion. E. Dautriat, a former director of the Launcher Directorate of the CNES recently declared: “Since the risk is inherent to any human activity, the question is to know how to discover it, grasp it, anticipate it, quantify it, and then take the corresponding decisions, in order not to suppress the risk – which is vain and which would sterilize any initiative – but to manage it.”

E. Dautriat continues: “Application of risk management to industrial processes and to products is not new. … It demonstrates its efficiency in the nuclear domain and in space activity in particular. It is of course from the very initial phase of design that a dependability approach should be applied; but at the origin it does not aim at controlling this designing process itself. However, a dependability approach should now take into account the developing process itself, being aware of the difficulty even greater … in the case of innovative projects."

These statements sustain the view that in a complex system such as large projects, risks deserve to be apprehended on a knowledge/information level.

Chance is therefore the consequence of a gap between the available information and the necessary information, which allow deciding the result of an experience. This gap has two origins: (a) the unavailability of information at a given moment because they are out of reach; (b) the complexity of the considered process or the number of pieces of information to be processed even if they are all available. This also covers the fastness of evolution of a process to reach a result.

The case of the *Rosetta mission* we describe below offers a perfect example of a risk management case where the delimitation of the available/necessary information domains as well as their evolution constitute a challenge for insuring is dependability.

Three types of risks are usually considered in risk management activities:

- *Company risks* which are related to the perenniality of the company
- *Project risks* which are related to (a) the performance of the product (which is targeted of the project), (b–c) the cost and time factors (for the project), (d) the safety of the product
- *Product risks* which is related to the exploitation of the product itself: its availability, safety

It is currently the two last types of risks that are considered in space activities however the *company risks* are analysed and managed from time to time within space companies or agencies such as CNES.

The Rosetta project

The ROSETTA Mission of the European Space Agency (ESA) will study comet Churyumov Gerasimenko with which the probe has a rendezvous in August 2014. (All the information about the Rosetta project can be found on the CNES site: www.cnes.fr in the "CNES programmes" entry, then "*Research and innovation*" entry then "*Rosetta*" sub-entry, or on ESA site: http://sci.esa.int/ entry "Satellites in orbit", then sub-entry "Rosetta status report".)

After a period during which a global mapping of the comet will be realized by the orbiter, a closer observation phase will follow, including the sending of a module (Lander) down to the comet.

The launch, that took place 2 March 2004 by an Ariane 5 launcher, will lead to a placing in the right orbit near the comet by August 2014 for an 18-month observation period.

The International Rosetta Mission was approved in November 1993 by ESA's Science Programme Committee as the Planetary Cornerstone Mission in ESA's long-term space science programme. The mission goal was initially set for a rendezvous with comet 46 P/Wirtanen. After postponement of the initial launch a new target was set: Comet 67 P/Churyumov-Gerasimenko. On its 10-year journey to the comet, the spacecraft will hopefully pass by at least one asteroid.

Few enterprises are more difficult or hazardous than space travel. Yet, even when compared with the achievements of its illustrious predecessors, ESA's Rosetta mission to orbit Comet Churyumov- Gerasimenko and deploy a lander on its pristine surface must be regarded as one of the most challenging ventures ever undertaken in more than four decades of space exploration.

Having overcome the time constraints associated with the launch, the hundreds of engineers and scientists involved in Rosetta are now about to face the ultimate assessment of their endeavour – the ability of their creation to not only survive in deep space for more than a decade, but to successfully operate in the close vicinity of a comet and return a treasure trove of data that will revolutionize our knowledge of these mysterious worlds. The suite of 21 scientific instruments on board Rosetta will return data on how a comet behaves in the outer reaches of the solar system and what happens as it gets closer to the Sun, and reveal the composition and structure of its nucleus.

Because of its long travel the question of the knowledge preservation becomes a critical issue both for the mission and for the different instruments designed by the scientists. Later the exploitation of the scientific data, five or even ten years after the end of the mission, will represent a new challenge for the scientific teams involved.

How therefore to manage the project risks (associated to the probe and the lander mission) and the product risks (associated to the instruments and the corresponding scientific data)?

3 Contributions of the Book

Five groups of contributions constitute the present book. The two first groups address the main topic of the workshop, i.e.

1. Management of Projects in their Early Phases
2. Risk Management

And three others deal with applications of organizational semiotics to philosophical, social, and technical issues were also considered, i.e.

1. Organizational Semiotics and Multi-Agent Paradigm
2. Transformation of Information
3. Application of Organizational Semiotics

3.1 Management of Projects in their Early Phases

A very large project in its formative stages tends to resist the application of established management tools and techniques that are better suited to later stages when the product is well understood, its manner of production established and the work – although complex – is subject to a stable plan of action.

Between stating the broad objectives and defining precisely the means of satisfying them with a clear plan of action, the project requires a rapidly growing and changing community that must also develop trusting relationships even while working on designs and plans that necessarily introduce many conflicting creative ideas. OS can contribute to the management of these early, turbulent processes, for example, our understanding of the problems, provide methods of observation or analysis, improve communication among participants, supply problem-solving techniques or support the application of information and communication technologies (ICT). Following issues illustrate the kind of contributions of OS:

- Maintaining and adjusting the balance between informal and formal ways of working.
- When formality, including IT support, is introduced, the flexibility to cope with frequent changes of requirements should not be lost.
- The broad statement of objectives must be translated into precise definitions of what must be done and how – so the creation of meaning plays a key role in these stages.
- Solutions to such problems as these call upon many disciplines, and this raises problems of mutual understanding.
- Innovation may entail using new terminology. To what extent? How do project teams achieve this and negotiate agreement?
- Necessarily, some terminology will be rather vague at the project's beginning and much of the effort will be devoted to making it precise enough to prescribe successful action. How can progress on this be facilitated, tested and retained?
- At every stage many options will be open, especially at the beginning. Alternative ideas will compete and this many generate personal

rivalries. Can a wealth of creative thinking be encouraged without inhibiting the growth of trust and the formation of good relationships?

- While arriving at solutions to the numerous problems encountered the process may display various pathologies (for example, “group think” is one of them, when an idea gains a momentum it does not deserve because the group appears to have reached a consensus that no one is willing to criticize). Even when a good solution has been negotiated, unexpected events, financial difficulties etc may call for a change of track. Such volatility cannot be avoided.
- Does OS offer any strategies that might be tested by using experience in current projects for observation, investigation, or experiment?

Chapters 1, 2 and 3 address some of these issues.

In Chapter 1, “Using Problem Articulation Method to Assist Planning and Management of Complex Projects”, Kecheng Liu, Lily Sun, and Simon Tan describe a Problem Articulation Method applied in planning a major project of Infrared Atmospheric Sounding Interferometer (IASI) in CNES. The techniques of the method, one of the OS methods assist modelling the project by articulating the entire project into manageable units and linking these units with interconnected collateral relationships. The model can then further be used to analyse the requirements of each unit and its contribution to, and impact on, the entire project. The requirement specifications produced by this process can guide the detailing of project activities, budget, and resources allocation.

In Chapter 2, “Omissions in Managing Knowledge in Innovation Processes or how to Handle Knowledge, Humans, and Tasks: a Semio-cognitive Approach”, Ruben S. Cijsouw, René J. Jorna, Gerhard Rakhorst, and Bart J. Verkerke claim that, in organizations, innovation is a long-lasting process which is difficult to manage. Innovation is characterised by the use of new (combinations of) knowledge. In the literature, they identify five serious omissions with respect to the management of knowledge in innovation processes, such as the difficulty to deal with the dynamics of knowledge and the lack of dealing with task dependencies between individuals. In order to repair these problems they introduce a cognitive framework in which knowledge content (domain) and type (the way knowledge is presented) are distinguished.

In Chapter 3, “Viewpoint-centred Methodology to Design Project/ Subcontract Cooperation Policies”, Pierre-Jean Charrel and Caroline Thierry present a methodology to improve management projects which involve a project/subcontractor relationship and a shared resource. The project is viewed as an information system where each actor’s main activity relies on accurate negotiation in order to succeed. The methodology is based (1) on the use of a discrete event based simulator which takes into account features of a cooperation policy as an input and (2) on a model of the negotiation. This model is built on a semiotic inspired notion of viewpoint An algorithm is sketched to design a cooperation policy: it relies upon an iterated process that feeds and manages the simulator along the negotiation taking into account the viewpoint-centred analysis of the results of past simulations.

3.2 Risk Management

Among the definitions of risk, "combination of the probability of an event and its consequences" [2], and "combination of the probability of damage and its effects" [3] call up that risk is related to knowledge available about the domain of activity.

OS may contribute to risk management by studying such following issues:

- Although the early planning, design, and development stages provided the greatest scope for risk management decisions, options are not closed even after the launch of a spacecraft. What can be done to enlarge the scope for risk management?
- The flight of a spacecraft has many phases with particular associated functions and so risks change (the trajectory may include periods in orbit around planets and moons). How might the semiotics aspects of these phases relate to the style of risk management?
- The principle benefits from a project, especially one-off projects such as Rosetta (cf. section 2.2) only emerge when the spacecraft completes its mission and the laboratories begin years of analysis when data return to Earth. Should risk management decisions take into account the values of these data-streams?
- As a spacecraft may never reach its destination, what other values can such projects yield? How can they be identified and assessed, and used in making risk management decisions?
- When should a project's values be assessed and how often re-assessed during its life? And by whom?
- A long-term project lasting from its proposal to completion (including data exploitation) will engage a changing population of hundreds of scientists and engineers. A good organisational memory is essential and it can only be formal and documented in part.
- Documentation and computer records will use terms with no guarantee that, over three decades, they will continue to represent the same concepts.

- Terminology at each point of time will be understood in the context of the current state of knowledge and the associated informal culture that provides its interpretation. How can we monitor relevant changes in knowledge and cultural context over 30 years and how should we react to them?
- Even a failed project should yield skills and knowledge, much of it informal: how can these be registered, evaluated and redeployed effectively?
- Informal systems play vital roles, especially during the project's turbulent early phases. How can they be made effective? How can their practices and achievements (such as creating new negotiated meanings) be anchored into the organisational memory?

In Chapter 4, "A Contribution to a Semiotic Approach of Risk Management", Daniel Galarreta examines how a semiotic approach of risks can be proposed and how the OS affordance concept can be adapted to such a goal. Perception issues are examined in order to make clear the relation between the concepts of action, here closely related to risk, and of affordance. A multi-viewpoints semiotics offers a convenient framework for defining a risk as a semiotic concept. In the case of the Rosetta long duration mission, it appears that managing risks of knowledge evolution, in order to prevent uncontrolled knowledge evolution, should be based on the combination of text-mining techniques and organisational arrangements.

3.3 Organizational Semiotics and Multi-Agent Paradigm

Chapters 5 and 6 address communication issues related to multi-agent paradigm where OS brings up new look.

In Chapter 5, "Norm-based Contract Net Protocol for Coordination in Multi-agent Systems", Juhua Wu and Renchu Gan study Contract Net Protocol (CNP), often used for coordination in a multi-agent system. Due to the limitations inherent in the conventional CNP, this paper proposes a Norm-based CNP to improve the efficiency and effectiveness of the coordination processes in a multi-agent system. Firstly, a three-dimensional taxonomy of norms is put forward in terms of the hierarchy, type, and flexibility of norms. Then a coordination process guided by Norm-based CNP is developed under the taxonomy framework. It provides a feasible solution for the optimization of the candidate selection, illustrated on a case study.

In Chapter 6, "Interaction of Simulated Actors with the Environment", Henk W.M. Gazendam explores the possibilities of an improvement of the interaction of an actor with its environment including other actors. This is done in the framework of a project aiming at a multi-actor simulation environment based on the ACT-R architecture. Traditional cognitive architectures like Soar and ACT-R lack of physical grounding and symbol grounding. In order to improve this situation, organisational semiotics offers concepts for the encoding of the environment in the form of affordance signs, social constructs, and social norms. This leads to new declarative chunk types in ACT-R. An emotion simulation subsystem is also presented which maintains an emotional state that encourages task performance, learning, and social behaviour. An awareness subsystem enables task switching based on the emotional state and the selection of those social constructs and norms that are applicable to the current situation.

3.4 Transformation of Information

One of the more addressed issues in OS is information transforming, especially in the field of Information System design.

In Chapter 7, “Semiotic Transformation from Business Domain to IT Domain in Information Systems Development”, Mingxin Gan, Kecheng Liu, and Botang Han propose a mechanism to transform business objects into Information Technology components. Semantic transaction loss exists in terms of concepts transformation from one design stage to another in information systems development. It results from different interpretations and representations of various requirements in design domains. In this paper, a mechanism for transformation connects different aspects of information systems with a precise and coherent representation. The transformation begins with the analysis of business objects in business domain, and finishes by generating corresponding structural components in IT domain. Components and their relationships in each domain are endowed with a correlated semantic interpretation. The processes of transformation are illustrated through signs and their structure in an OS perspective.

In Chapter 8, “Comparative Analysis of Ontology Charts and other Modelling Techniques”, José Cordeiro and Joaquim Filipe use the OS Ontology Charting (OC) technique to represent the requirements of organisational information systems, incorporating technical and social aspects. This chapter presents a comparative analysis of the modelling techniques used by some Information Systems designing methods, applied to a case study, discussing each model’s characteristics and expressive power. A framework to guide this comparison is also introduced enlightened by Semiotics.

In Chapter 9, “The Separation of Data and Information in Database Systems under an Organisational Semiotics Framework”, Xi Wang and Junkang Feng present a semiotic-based perspective to distinguish between data and information. Some significant problems in database systems research, such as query answering capability, connection traps, lossless transformation, and normalization are difficult to explain, answer, solve, or explore further within the current context where data and information are fused, or even taken as the same thing. OS enables to distinguish data and information, and takes data as a type of sign, which carry information. The authors look at how and why the contemporary seemingly muddled view on the relationship between data and information might have hampered the progress on a number of database research issues.

In Chapter 10, “Towards a Social-based Process for Information System Development: A Case Study”, Carlos A. Cocozza Simoni, Amanda Meincke Melo, and Maria Cecília C. Baranauskas contribute to understand organizations, their interactions, and their evolution. The role played by the computer in organizations continues to evolve and increases in importance, since it mediates social relationships. To improve the information system development process we need a better understanding of the organizations and their internal and external interactions and dynamics. This chapter discusses a semiotic-based approach to the development of information systems. It is illustrated with a case study in which a real organization was exposed to methods of OS to rethink its way of developing systems.

3.5 Applications of Organizational Semiotics

The last chapters present applications of Organizational Semiotics.

In Chapter 11, “A Semiotic Framework for Research into Self-Configuring Computer Networks”, John H. Connolly, Iain W. Phillips, Lezan Hawizy, and José Ignacio Rendo-Fernández focus on communication process to improve network configuration. Self-configuring computer networks are designed to offer services to users in response to their specific requirements on particular occasions. In order for such networks to obtain information about their users’ requirements and then to respond appropriately, processes of communication need to take place between the user and the network, and within the network itself for the purpose both of configuring the network appropriately and of providing the required services. These processes can be analysed in terms of a multi-level semiotic framework in such a way as to clarify the understanding of their properties in relation to structure, meaning, and contextually situated use.

In Chapter 12, “The Semiotics of Usage-Centred Design”, Jennifer Ferreira, James Noble, and Robert Biddle claim that a user interface is well designed when designers have correctly mapped the application domain onto the solution domain. This mapping may be helped by the design methodology and the success of any software engineering methodology depending on the mapping it provides between the application domain and the solution domain. A good match between the requirements and the implementation reduces the risk of having to make costly and major changes to the user interface at a late stage in development. This chapter uses semiotics to provide a better understanding of the models and the process of the Usage-Centred Design methodology so as to understand its success.

Programme Committee

Peter Bøgh Andersen, Aalborg University, Denmark
Cecilia Baranauskas, University of Campinas UNICAMP, Brazil
Pierre-Jean Charrel (co-chair), Universisté Toulouse 2 Mirail, France
Rodney Clarke, Staffordshire Univesity, United Kingdom
Samuel Chong, Manchester Univesity, United Kingdom
Joaquim Filipe, Escola Superior de Tecnologia de Setúbal, Portugal
Daniel Galarreta (co-chair), Centre National d’Etudes Spatiales, France
Henk Gazendam, Groningen University and Twente University, The Netherlands
Goran Goldkuhl, Linköping University, Sweden
Ricardo Ribeiro Gudwin, University of Campinas UNICAMP, Brazil
Rene Jorna (co-chair), Groningen University, The Netherlands
Kecheng Liu, The University of Reading, United Kingdom
Ronald Stamper, Twente University, The Netherlands

References

1. Gazendam HWM, Jorna RJ, Liu K (2004) Round Table Workshop “An organizational semiotic view on interculturality and globalization”. Proceedings of the IASS 2004 Conference (2004), Lyon, France, July
2. ISO/IEC (2002) Risk management – Vocabulary – Guidelines for use in standards. Guide 73
3. ISO/IEC (1999) Safety aspects – Guidelines for their inclusion in standards. Guide 51

Contributors

Baranauskas, M.C.C.
IC – Institute of Computing
Unicamp, State University of Campinas
Caixa Postal 6176
13084-971, Campinas, SP
Brazil

Biddle, R.
Human-Oriented Technology Laboratory
Carleton University
Ottawa
Canada

Charrel, P.J.
Université Toulouse 2 Le Mirail
Département Mathématiques et Informatique – GRIMM Research Team
5 Allée Antonio Machado
F-31058 Toulouse Cedex 9
France

Cijsouw, R.S.
Faculty of Management and Organization
University of Groningen
PO Box 800
9700 AV, Groningen
The Netherlands

Cocozza Simoni, C.A.
IC – Institute of Computing
Unicamp – State University of Campinas
Caixa Postal 6176
13084-971, Campinas, SP
Brazil

Connolly, J.H.
Department of Computer Science, Loughborough University
Loughborough, LE11 3TU
United Kingdom

Cordeiro, J.
Escola Superior de Tecnologia de Setúbal
Rua do Vale de Chaves, Estefanilha
2910-761 Setúbal
Portugal

Feng, J.
Database Research Group
School of Computing
University of Paisley
PA1 2BE
United Kingdom

Ferreira, J.
School of Mathematics, Statistics and Computer Science
Victoria University of Wellington
P.O. Box 600
Wellington
New Zealand

Filipe, J.
Escola Superior de Tecnologia de Setúbal
Rua do Vale de Chaves
Estefanilha, 2910-761 Setúbal
Portugal

Galarreta, D.
Centre National d'Etudes Spatiales
18, av. Edouard Belin
31401, Toulouse cedex 9
France

Gan, M.
School of Management and Economics
Beijing Institute of Technology
Beijing 100081
China

Gan, R.
School of Management and Economics
Beijing Institute of Technology
Beijing 100081
China

Gazendam, H.W.M.
Groningen University & Twente University
P.O. Box 800
NL-9700-AV Groningen
The Netherlands

Han, B.
School of Management and Economics
Beijing Institute of Technology
Beijing 100081
China

Hawizy, L.
Department of Computer Science
Loughborough University
Loughborough, LE11 3TU
United Kingdom

Meincke Melo, A.
IC - Institute of Computing
Unicamp, State University of Campinas
Caixa Postal 6176
13084-971, Campinas, SP
Brazil

Noble, J.
School of Mathematics, Statistics and Computer Science
Victoria University of Wellington
P.O. Box 600
Wellington
New Zealand

Phillips, I.W.
Department of Computer Science
Loughborough University
Loughborough, LE11 3TU
United Kingdom

Rakhorst, G.
University Medical Centre Groningen
University of Groningen
A. Deusinglaan 1, 9713 AV, Groningen
The Netherlands

Rendo-Fernández, J.I.
Department of Computer Science
Loughborough University
Loughborough, LE11 3TU
United Kingdom

Sun, L.
Informatics Research Centre
The University of Reading
Whiteknights, Reading, RG6 6AY
United Kingdom

Tan, S.
Informatics Research Centre
The University of Reading
Whiteknights, Reading, RG6 6AY
United Kingdom

Thierry, C.
Université Toulouse 2 Le Mirail
Département Mathématiques et Informatique & ONERA-CERT, DCSD
5 Allée Antonio Machado
F-31058 Toulouse Cedex 9
France

Verkerke, G.J.
University Medical Centre Groningen & University of Groningen
A. Deusinglaan 1
9713 AV, Groningen
The Netherlands

Wang, X.
Database Research Group
School of Computing
University of Paisley
PA1 2BE
United Kingdom

Wu, J.
School of Management and Economics
Beijing Institute of Technology
Beijing 100081
China

Part 1
Management of Projects in their Early Phase

Chapter 1
Using Problem Articulation Method to Assist Planning and Management of Complex Projects

K. Liu, L. Sun, S. Tan

Informatics Research Centre, The University of Reading, Whiteknights, Reading, RG6 6AY, UK

Abstract

The planning of complex projects involves organising infrastructure and resources, analysing stakeholders and their responsibilities, and defining deliverables. The outcomes of this process may impact strategically on the success of the project which should deliver business values. In this chapter, we describe a problem articulation method (PAM) applied in planning a major project of infrared atmospheric sounding interferometer (IASI) in CNES. The techniques of the method assist modelling the project by articulating the entire project into manageable units and linking these units with interconnected collateral relationships. The model can then further be used to analyse the requirements of each unit and its contribution to and impact on the entire project. The requirement specifications produced by this process can guide the detailing of project activities, budget, and resources allocation.

Keywords: project planning, project management, organisational semiotics, project planning requirements specifications

1.1 Introduction

Planning of complex projects is a challenging process that must ensure an alignment between requirements for the project development and

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 3–13.

requirements for the project planning (Liu *et al*. 2002). It is thus imperative to establish a holistic view of all units of the project as well as interactions and communications between them. A large complex project requires an effective method for capturing the requirements of project planning in relation to policies, constraints, assumptions, and processes which should be transparent to all the stakeholders (Liu 2000). When there are changes in the project, the project management should be able to respond and adjust the change effectively towards the success of the project.

Complex projects tend to be late in completion, over budget, and often resulting in poor quality systems (Bounds 1998). Solutions to this problem have included the development of formal approaches to software process improvement (Herbsleb *et al.* 1994) and the application of formalised project management methods to plan, monitor, and control budget, time, and quality. Recent efforts have begun to integrate software process improvement methods with more generic project management methods (Pennypacker and Grant 2003). However, these methods are incapable of analysing the requirements prior to planning the operation of the project that may lead to limited understanding of the project and subsequently inadequate planning of the project. In order to avoid this type of risk, we apply problem articulation method (PAM) (Stamper and Kolkman 1991; Kolkman 1993; Stamper 2001) to assist project planning and management.

PAM is a method which articulates and decomposes complex problem situations into manageable units and their interconnected relationships. A focal unit system is referred to as the key objective to achieve while other unit systems serve as the infrastructure within the whole context. PAM is suitable for analysis and design of enterprise and IT applications (Stamper *et al.* 2004), which provides and facilitates cost-benefit analysis, project management, and project planning.

PAM is comprised of five techniques: unit systems definition, stakeholder analysis, collateral structuring, organisational containment, and valuation framing, which assist the process of articulation, analysis, and planning for projects. The technique of *unit systems definition* breaks down the complex project, as a problem situation, into manageable components which are defined as unit systems. One of the unit systems will be further considered as a focal system. The rest of the unit systems are considered as collateral systems, some of which provide services to the focal system. An important part of the analysis for the complex project is to describe stakeholders with their roles and responsibilities. The technique of *stake-holder* *analysis* enables to document the stakeholders involved in conjunction with the unit systems. Their participation can be valued by the feedback from the stakeholders' viewpoint to reveal the impact and their perceived value of the project. To illustrate PAM, we use the IASI project (Galarreta 2003) as a case study. The project requirements are analysed and specified for the project planning and management.

1.2 Articulation of Requirements of the IASI Project for Planning

The IASI project sets its aim at producing three flight models, data processing software, and a technical expertise centre. This project involves a number of stakeholders, such as development teams from Centre National d'Études Spatiales (CNES), EUMETSAT, production teams from Alcatel Space, and other end-users. The project complexity lies in the multiple suppliers, interrelated activities and the stretched project life span.

1.2.1 Capturing the requirements of the IASI project

In order to plan efficiently the IASI project, we apply PAM to holistically describe what this project involves in terms of related development of devices and software. The planning and management of the project can then take the requirements of the project as the basis to identify and allocate the resources.

The technique of unit systems identification in PAM allows us to break the entire project down to the related unit systems as shown in Fig. 1. The criteria we adopt for identifying these unit systems are:

A unit system normally consists of a collection of organised activities performed by people or automata to achieve a set of objectives.

The analysis of each unit and its subunits is a recursive process. The requirements of IASI can be articulated until all the related units and their relationships within the entire project are holistically described to all stakeholders. The technique of *unit system definition* provides an effective means of examining the problems situation with information provided by the relevant domain experts to clearly define the problems, tasks, and plan of actions.

```
<unit systems>
   U1 = IASI device in operation in Metop satel-
lites
   U2 = daily management
         <sub-unit systems>
            U2.1 = documentation management
            U2.2 = delay management
            U2.3 = action management
            U2.4 = configuration control
            U2.5 = risk management
         </sub-unit systems>
   U3 = Technical Expertise Centre
   U4 = Commission of IASI
   U5 = CNES software development
         <sub-unit systems>
            U5.1 = data processing software
         </sub-unit systems>
</unit systems>
```

Fig. 1. Description of the unit systems in the IASI project

1.2.2 Stakeholders and their roles in the project

Stakeholders normally have impact on the outcomes of the project. Mennecke and Bradley (1997) argued that a project with clearly assigned roles and responsibilities produced higher quality deliverables than those without. It, therefore, is important that stakeholders' roles and responsibilities involved in the project must be clearly identified and incorporated in the planning. PAM recognises six roles of stakeholders which can be referred to in describing their responsibilities in the IASI project.

$U \supset$ {Stakeholder, Role, Responsibility}
where
Role = Actor|Clients|Provider|Facilitator|GoverningBody|Bystander

- *Actor*: An actor has a direct impact on the action course. This role often involves substantive and message passing amongst the other roles.
- *Client*: A client is the user or beneficiary who is the recipient of the consequences or outcome of U.
- *Provider*: A provider is the developer who is responsible for creating the conditions and resources to facilitate the deliverable

of a U, e.g. supplies and authorisation to enable the functionality and operability of the project.

– *Facilitator*: A facilitator is the initiator and enabler of the system and acts as a focus point for the group in directing the action course. A primary responsibility of a facilitator is to resolve conflicts and ensure continuality, steering the team towards the organisational goal.
– *Governing body*: Determines the strategic aims, high-level objectives and direction of the U to keep them under review and ensuring they are on-track. A governing body may take part in the management planning for the U, e.g. goals, budget, and partners' collaboration.
– *Bystander*: Bystanders exert a participant role of shaping the action course. The *bystanders* are usually not part of the project itself. However, bystander will influence the system and its outcome in many ways determining the course of the project.

Based on the unit systems described in Fig. 1, the stakeholders are identified as detailed in Table 1. In this table, the columns capture the information for various job functions and responsibilities within the corresponding unit systems which are represented by the roles. The outcome from this stage of analysis can indicate the activities within which the stakeholders are responsible.

Table 1. Description of roles and responsibilities within the IASI project

Unit system	Stakeholders	Roles	Responsibility
U_1 IASI device in operation in Metop satellites	SH_1 EUMETSAT Engineers Analysts Programmers	Facilitator (collaborators)	Provide requirements to the CNES' development team; provide the system to clients
	SH_3 CNES: Engineers Analysts Developers Project managers	Provider and facilitator	Lead the development of the IASI project; cost sharing agreement, payments, and price revision; coordinate and manage all the suppliers
	SH_4 Apcatel space	Provider	Design and assemble three IASI instruments

(contd)

Table 1. (contd)

Unit system	Stakeholders	Roles	Responsibility
U_2 Daily management	SH_1 CNES Project managers	Facilitator	Manage day-to-day operations
$U_{2.1}$ Documentation management	SH_1 CNES Project manager	Facilitator	Oversee and coordinate documents collection, filing, distribution
	SH_2 CNES Clerks	Actor	Record and maintain documents
$U_{2.2}$ Delay management	SH_2 CNES Project manager Team leaders	Facilitator	oversee and coordinate all parts of development; liaise with the suppliers
$U_{2.3}$ Action management	SH_1 EUMETSAT: Technicians Operators Engineers	Client	monitor the progress coordinate project activities
$U_{2.4}$ Configuration control	SH_1 CNES and EUMETSAT Engineers Technicians Developers	Actor	Configure the system post to installation; technical trouble shooting
	SH_2 CNES and EUMETSAT Project managers	Facilitator	Oversee and coordinate operations between "provider and customers"
$U_{2.5}$ Risk management	SH_3 CNES and EUMETSAT Project managers	Actor	Identify and appraise the risks; prioritise the alternatives and make decisions
U_3 Technical expertise centre	SH_1 EUMETSAT Operator Consultants	Actor	Request services; aid the clients for information processing.
	SH_2 CNES Analysts Developers Project manager	Provider	Develop the technical expertise centre; operate the technical expertise centre

Table 1. (end). Description of roles and responsibilities within the IASI project

U_4 Commission of IASI	$\boxed{SH_1}$ EUMETSAT Project manager	Actor	Negotiate and make agreement for the project; negotiate and communicate for requirements for the development
	$\boxed{SH_2}$ CNES Analysts Project manager	Provider	Propose and execute the development commission of IASI project
	$\boxed{SH_3}$ Apcatel Space	Provider	Design and assemble three IASI instruments
U_5 CNES software development	$\boxed{SH_1}$ CNES Analysts Developers Project manager	Provider	Develop application software
$U_{5.1}$ Data processing software	$\boxed{SH_1}$ EUMETSAT Engineers Clients	Provider	Provide requirements communicate the design test the applications
	$\boxed{SH_2}$ CNES Analyst Developers Project manager	Actor	Design and implement the application test the application provide maintenance of the application through the technical expertise centre

1.2.3 A collateral structuring model for the IASI project planning

Once all the unit systems and their corresponding stakeholders' involvements in the IASI project are articulated, a coherent view can be established by integrating the contributing activities indicated in the unit systems. We use the techniques of collateral structuring model to visualise an integrated view of the IASI project in Fig. 2. Two types of systems are identified: the object systems (in rectangles) and service systems (in ellipses). The latter provide services and operations on the former.

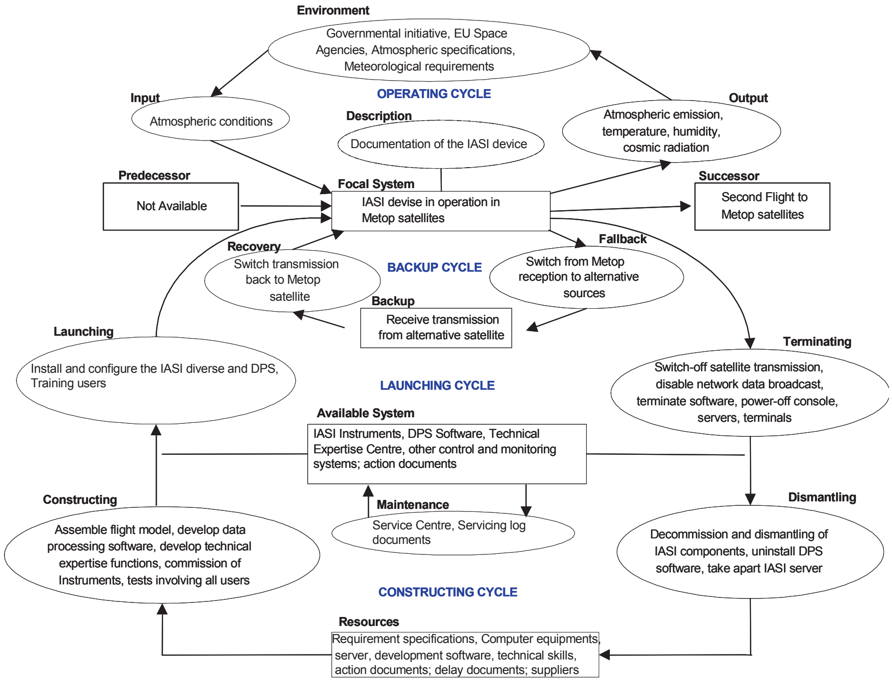

Fig. 2. A collateral structuring model of the IASI project

One of the unit systems in Fig. 1 can be selected as the focal system which represents the particular focal interest of the project. The chosen unit systems are surrounded by the service systems, e.g. constructing system, launching system, maintenance system, disseminating system, and terminating system; and other object systems such as backup system and available resources system. These collateral systems fall into cycles: constructing, launching, operating, and backup, which provide a mechanism for architecting the IASI project. For example, the construction system in the constructing cycle involves several activities from $U_{2.3}$, U_3, U_4, and $U_{5.1}$ and requires input from resources. These resources must be available before the construction can take place, e.g. requirement specifications provided by $U_1(SH_1, SH_2)$; computer equipments provided by $U_1(SH_3)$ and indirectly by $U_1(SH_4)$; and server, development software, technical skills, action documents, delay documents provided by $U_1(SH_3)$. The next stage of the project moves on to launching the system, which involves the installation and configuration of the IASI diverse and DPS, and providing training to all relevant users.

The collateral structuring model can be used to document the entire project with the information which is required by planning and management of the IASI project.

1.3 Requirements Specifications for Planning the IASI Project

A PAM tool based on the collateral structuring model has been developed for project planning. This tool enables the planning focusing on each individual unit system with its required resources for the time and cost involved by carrying out all the activities, and then aggregating automatically the overall resources for the entire IASI project.

Figure 3 presents the requirement specifications in the interface of the software tool where the key information for the project planning can be inputted and the resources required can be computed.

The use of the PAM method has demonstrated that the IASI project can be viewed holistically through its objects systems with the associated service systems which contain the detailed level of information for the different contributing parts to the overall project, and produce the requirements for planning the IASI project. Some of the information in the requirements, such as the duration, and status of the resource consumption, can be monitored to support the project management and decision-making.

1.4 Conclusions

Empirical studies in project planning of complex systems have shown the importance of social-technical, business, and organisational issues for successful implementation. Although software engineering methods have been embedded in the project planning process that ensures thorough understanding of the project being planned, but holistically describing the complex project and planning it still remains challenging. The PAM method provides the mechanism to resolve this challenging problem. The techniques in PAM produce a dynamic view of the IASI project. The outcomes from the collateral structuring model can automatically generate the requirements specifications for the project planning and management respectively. In this chapter, the requirements specifications for planning the IASI project have been presented and the real-time refinement for project control is performed during the IASI project's implementation.

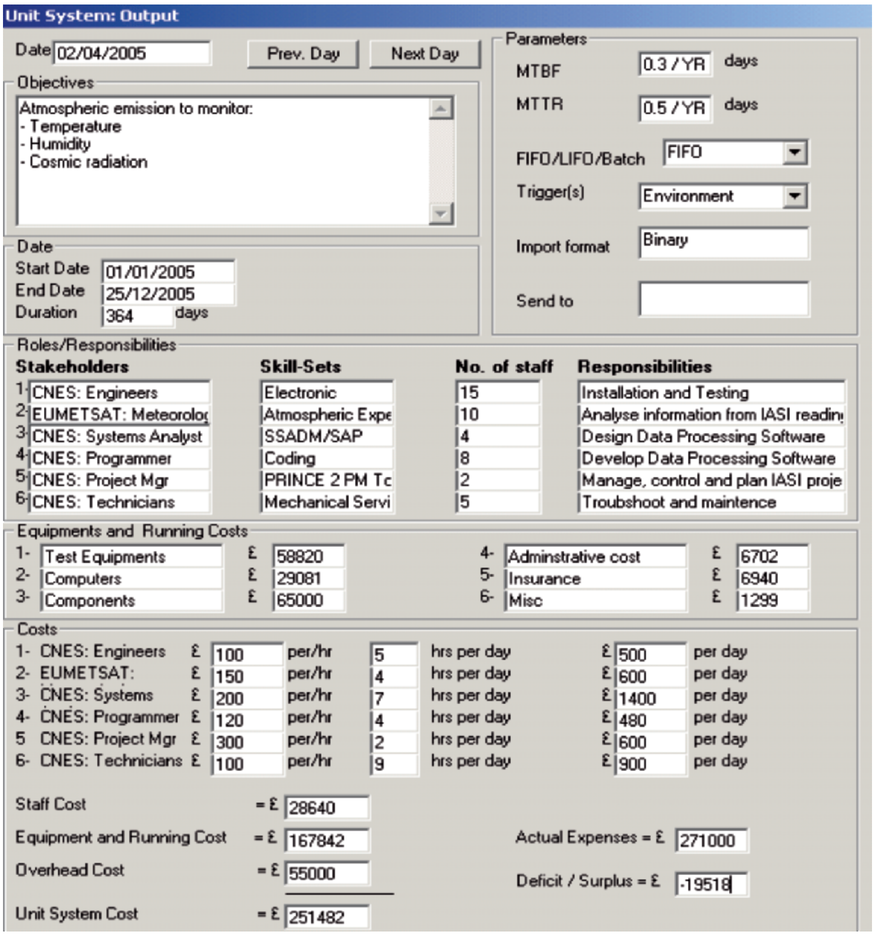

Fig. 3. Unit systems template

Acknowledgement

This research is partly supported by EPSRC – SEDITA project GR/S04840/01.

References

Bounds S (1998) The last word on project management. IEE Solutions, 30(11), pp. 41–44

Galarreta D (2003) Designing Space Systems in Multi-Viewpoints Semiotics, Proceedings of the 6th International Workshop on Organisational Semiotics, The University of Reading, Kluwer, UK

Herbsleb J, Carleton A, Rozum J, Siegel J, Zubrow D (1994) Benefits of CMM-Based Software Process Improvement: Executive Summary of Initial Results, Carnegie-Mellon Software Engineering Institute, Pittsburgh

Kolkman M (1993) Problem articulation methodology, PhD thesis, University of Twente, Enschede

Liu K (2000) Semiotics in Information Systems Engineering, Cambridge University Press, Cambridge, UK

Liu K, Sun L, Bennett K (2002) Co-design of business and IT systems. Information Systems Frontiers, 4(3), pp. 251–256

Mennecke B, Bradley J (1997) Making Project Groups Work: The Impact of Structuring Group Roles on the Performance and Perception of IS Project Teams. International Academy for Information Management, 12th Annual Conference, pp. 19–24

Pennypacker J, Grant K (2003) Project management maturity: an industry benchmark. Project Management Journal, 34(1), pp. 4–11

Stamper RK (2001) Organizational semiotics, evolving science of information systems. In: Liu K, Clarke R, Andersen PB, Stamper R (eds.), Kluwer Academic, Boston

Stamper RK, Kolkman M (1991) Problem articulation: a sharper-edged soft systems approach. Journal of Applied Systems Analysis, 18, pp. 69–76

Stamper RK, Liu K, Sun L, Tan S, Shah H, Sharp B, Dong D (2004) Semiotic Methods for Enterprise Design and IT Applications, Proceedings of the 7th International Workshop on Organisational Semiotics, Portugal, Escola Superior Tecnologia, INSTICC Press

Chapter 2
Omissions in Managing Knowledge in Innovation Processes or How to Handle Knowledge, Humans and Tasks: A Semio-Cognitive Approach

R.S. Cijsouw[1], R.J. Jorna[1], G. Rakhorst[2], G.J. Verkerke[2]

[1]*Faculty of Management and Organization, University of Groningen, PO Box 800, 9700 AV, Groningen, The Netherlands*
[2]*University Medical Centre Groningen, University of Groningen, A. Deusinglaan 1, 9713 AV, Groningen, The Netherlands*

Abstract

In organizations, innovation is a long-lasting process that is difficult to manage. Innovation is characterized by the use of new (combinations of) knowledge. Innovation, as knowledge creation, is also an activity of individuals. However, neither the individual nor knowledge is studied as appropriate unit of analysis in innovation and knowledge management literature. In this chapter, we start with two cases from the literature that indicate problems with respect to knowledge in innovation projects. In a more fundamental review of the literature, we identify five serious omissions with respect to the management of knowledge in innovation processes, such as the difficulty to deal with the dynamics of knowledge and the lack of dealing with task dependencies between individuals. In order to repair these problems, we introduce a cognitive framework in which knowledge content (domain) and type (the way knowledge is presented) are distinguished. In the conclusion, we benchmark the cognitive framework with the current methods using the five omissions as guidelines. This contribution is analytical, diagnostic, and conceptual. In the conclusion a framework is designed that is empirically tested in various innovation projects (Cijsouw 2006).

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 15–45.

Keywords: knowledge management, innovation, semio-cognitive framework, project management.

2.1 Introduction: Motive and Structure

2.1.1 Motive and relevance

Knowledge is an important resource in business processes, especially in innovation and research and development (R&D). Compared to most "routine-based", or repetitive, processes, innovation itself is a business process with a high level of uncertainty. In innovation processes, uncertainty may negatively influence the ability to realize the desired output, the process duration, the necessary input, and the architecture of the innovation process. These difficulties are often visible in long throughput times, in changes of staff members or task roles, in redoing knowledge activities, in gaps in task and knowledge connections, and in inabilities to coordinate and plan. We argue that these negative aspects of innovations result from negligence of fundamental knowledge dynamics at the level of description of the individual.

Innovation at the organizational level is characterized by the use of new (combinations of) knowledge. For this reason, the dynamics of knowledge is of interest for the management of innovation processes. That is, it is important to manage the creation and transfer of knowledge in innovation contexts.

At a lower level of description, individuals always create knowledge. Without human cognition, there is no knowledge creation. We take this individual perspective for granted. However, the success of knowledge creation in innovation is realized by the cooperation of individuals, especially when after the invention the implementation phase has to be realized. Therefore, we include the individual as well as the group level in our analyses, but we consider individuals to be the lowest (ontological) level in these organizational discussions. Therefore, it is surprising that the individual is not also the unit of analysis in old and new literature on innovation management, project management, and even knowledge management (e.g. Brown and Eisenhardt 1995). Instead, teams or organizations are the units of analysis in this literature. To put it differently, old and current literature on innovation and knowledge management does not use the individual level – including the tasks individuals perform – to describe, analyse, and determine what is going on in innovation processes.

A remark of caution is necessary. The literature on innovation and R&D does not completely neglect knowledge (e.g. Tijssen 2001; Frederiksen, Hemlin, and Husted 2004). It addresses various knowledge-related aspects that could influence management policy and activities: such as knowledge utilization (Landry, Amara, and Lamari 2001; Frederiksen *et al.* 2004), knowledge sharing (Meyer-Krahmer and Schmoch 1998; Tezuda and Niwa 2004), knowledge transfer (Zellner 2003; Ernst and Kim 2002; Pisano 1996; Szulanski 2000), the commercialization of intellectual property (Goldfarb and Henrekson 2003; Rappert, Webster, and Charles 1999), and the network position of the organization and persons (Barras 1990; Evaristo *et al*. 2004; Kazanjian, Drazin, and Glynn 2000). Many of these aspects influence each other. For instance, knowledge sharing influences knowledge utilization. However, knowledge-related aspects are not integrated into innovation management.

We wonder whether sufficient insight in the dynamics of knowledge can be obtained, and hence in the innovation processes itself, if the individual level is not the primary unit of analysis. The use of (ontological) higher levels of analysis, such as teams and organizations, is reflected in the top-down approaches in innovation management. In contradistinction, we argue that innovation processes can only be managed using a bottom-up approach, that is if the individual also is a unit of analysis. For this reason, we will construct a framework that looks at the individual as an important unit of analysis. This framework is used to manage, observe, analyse, and determine knowledge in innovation processes (Cijsouw 2006). The framework also takes the dynamics of knowledge into account.

2.1.2 Aim and structure

In this chapter, we describe negative consequences of managing innovation processes result from neglecting innovation processes looked at from the point of view of the individual level of analysis. One important negative consequence is the inability to manage dynamics of knowledge in innovation processes resulting in gaps in knowledge and task dependencies and in longer duration times of innovation projects. We argue that if one is not able to manage knowledge aspects at the adequate level of description, one is not able to positively influence the results. To illustrate this point in innovation projects, we first describe two case studies (section 2). Then we review the innovation management, project management, and knowledge management literature (section 3) to see whether the case studies are exceptions. After discussing the case studies and the literature, we establish several omissions and as a follow-up, we propose a framework that makes it possible to manage the dynamics of knowledge in innovation processes (section 4). This framework starts with the individual (and his tasks) as the most important unit of analysis. Furthermore, the framework takes into account the longitudinal aspect of the innovation process. For instance, the different tasks a person has to execute during the total project time and the knowledge he uses and possibly creates to carry out these tasks. The framework is intended to be additional to current innovation, project, and knowledge management methods and tools. We believe it supports management's activities and policies in innovation and R&D processes better than the current method and tools. In the conclusion (section 5), we discuss the value of the proposed new framework and refer to ongoing empirical research in medical devices research (Cijsouw 2006).

2.2 Illustration of the Motive: 2 Case Studies

This section describes two case studies to illustrate the problems in the management of knowledge in innovation processes. The first case study is derived from empirical work that was carried out in the Minnesota Innovation Research Project (Van de Ven 1999). This project is an example of the most recent perspective in the innovation literature, the interactive process perspective. The second case study is derived from knowledge management literature. Unlike much of the knowledge management literature – which pays attention mostly to existing, repetitive processes – Swan *et al.* (Swan 2003) used this case study to link knowledge management to innovation.

2.2.1 Case study I: development of cochlear implants at 3M

Summary of the case study description

In 1977, the American company 3M became involved in the cochlear implant development. 3M aimed to successfully develop – that is commercially viable – cochlear implants. Initially, they cooperated with three university research groups. In this early stage, 3M decided to strive to become the first mover in the cochlear implant market. 3M first developed the technology that could be easily realized (the single channel device) in order to have the highest chance of actually being the first provider of cochlear implants on the market.

3M became the first company on the cochlear implant market once the Food and Drug Agency (FDA) was convinced of the device's safety – necessary for market introduction. To be able to convince FDA, 3M's development team had to transfer knowledge, mostly in the form of documents, i.e. coded knowledge (see section 4), to the FDA. This knowledge was related, among others, to audiological tests.

3M decided that the development of the next generation (the multichannel device) would follow the market entry of the single channel devices. However, the development of the single channel device took longer than initially estimated. In addition, sales of the single channel cochlear implant were smaller than 3M expected. The causes for the delay in the development of the single channel device are not clearly specified in the case study. The longer development and market introduction period of the single channel device caused the need for extra resources. These resources could not be used to develop the second-generation cochlear devices. This explains 3M's late start with the development of multichannel devices. Due to this late start, 3M was unable to compete to be the first mover of second generation cochlear implants as well.

Although 3M realized a device, it did not gain the market share it had expected from first generation devices. Sales were negatively influenced by market expectations – of patients and physicians – on the multichannel cochlear implants. Some were afraid of sustained damage, whereas others expected a radical performance increase in the next device generation. In the beginning of the project, this knowledge was not known in all departments of 3M. Performance increase of the second generation in comparison to the first generation was advertised by 3M's competitors and supported by FDA statements.

In 1988, 3M decided to exit the cochlear implants market because even minimal operations required substantial resources. These operations were distracting program members' attention from hearing aid-related activities (3M's new target market). The FDA allowed them to exit the market only after customer service and maintenance were secured.

Omissions in the 3M case

The case study gives insights in the maneuvering of a company to introduce a medical device in the market. The dynamics and particularities of the industry are used to explain certain events in the development process.

In the case study Van de Ven (1999) did not describe the phases within the R&D process, the tasks carried out, the individuals who carried out these tasks nor the knowledge that was needed to carry out the tasks. Not individuals, but the organization and the team were used as main units of analysis. Only occasionally, the development of the cochlear implant is described using individual roles. The university researchers were the only persons whose names have been mentioned and the senior managers are the only other individuals whose actions are explicitly mentioned; however not by name but by role.

The case study description does not include knowledge utilization and creation by individuals. The beginnings of the project, the early knowledge exchange and the dynamics in the early phases are not mentioned. There is also no reference to detailed knowledge content and knowledge type (to be explained in section 4). How the 3M engineers created and applied the cochlear implants is not mentioned in the case study. Of course, much knowledge is documented in the FDA protocols, but that is only end result knowledge. The case study only describes activities of knowledge transfer when FDA had to judge 3M's Pre Market Approval Application, that is, when 3M staff transferred necessary knowledge regarding testing, safety and performance in the form of documents and protocols to the FDA.

2.2.2 Case study II: knowledge management at British Telecom (BT) industries

Summary of the case study description

The second case study concerns the design and implementation of an integrated management information and planning system at BT industries (Swan 2003). The so-called sales support project (SSP) aimed at the design and implementation of an integrated management information and planning system. The system should be implemented in all European businesses of BT industries through the introduction of common, integrated IT platforms and information systems. Intsoft, a Swedish software supplier, designed and developed the software jointly with BT personnel.

The innovation project at BT industries was successful: the planning was late by only 1 month and with a few exceptions, it delivered the functionality that was needed. Furthermore, the long-term relationship with the software supplier was fine and high satisfaction and low turnover of project team and key users were reported.

A key knowledge management issue in this project was to identify those people at each division that had the relevant expertise and the interest and motivation to manage the implementation. The project teams consisted of divisional staff with detailed knowledge of local operating procedures, rather than knowledge in the IT domain. Teams were composed to comprise different "personality types". Two types of teams were distinguished: (1) design and development teams and (2) implementation teams.

In the design and development phase, Intsoft consultants worked together with BT managers, representing different functional areas and different European divisions, and with two (later four) newly employed graduates in

Business and IT. The graduates fulfilled roles as "knowledge brokers", because they worked partly on site at BT and partly in Intsoft. For a 12-week period and approximately 3 days a week these persons were brought together on one site in Sweden.

The implementation teams comprised representatives from Intsoft, BT's corporate IT function, divisional business managers, and representatives of each social community that would be affected by the system. The representatives from social communities had important knowledge of the local operating context. Little formal project documentation existed to transfer knowledge about the system from the implementation team to the users. This was mostly supported by verbal communication. Knowledge sharing was characterized by informal networking.

Omissions in the BT industries case

Knowledge management was extremely difficult in the BT industries case, because the management had no accurate insight in (1) the relevant communities, (2) the content of the knowledge that should be utilized, and (3) the way the created and utilized knowledge was presented and retained.

In the BT case study, the management of knowledge was most prominent in the composition of the development and implementation teams. The intention was to compose teams such that the relevant communities were represented in the team. However, in this innovation process, it proved to be difficult to identify the relevant community (who would use the innovation). The relevant knowledge content depended on the community that would be involved. For this reason, it was difficult to determine the relevant knowledge content beforehand (ex ante). This is a symptom of the fact that the unit of analysis is at a high level. It was the team instead of the individual. The uncertainty with respect to the composition of the innovation teams – and the identification of relevant communities – also had the effect that most knowledge management decisions were taken ad hoc: no reliable knowledge management in terms of knowledge content and type and involved individuals could be made.

The management of the innovation project could not take into account the presentation of knowledge, because it was difficult to conclude which knowledge and of what type had been created and utilized to carry out a certain task in the innovation process (Pierce and Delbecq 1977; Slappendel 1996). Therefore, management decided to bring together the dispersed personnel for 3 days during a 12-week period. In our opinion, the management could not determine whether the face-to-face meetings (sensory or implicit knowledge (to be explained in section 4)) were necessary.

If persons only used coded knowledge, knowledge transfer would also have been possible without these face-to-face meetings.

2.2.3 Highlights from the 3M and BT cases

In the two case studies, the most important units of analysis are the team and the organization. BT industries looked at the composition of the teams. However, the individual and the tasks were not described. In 3M, the documents indicated that knowledge was explicitly addressed only a few times.

We believe that in the case studies the management of innovation processes could have better governed the innovation, if they had known the knowledge content and appearance of the involved individuals. However, both the content and the types of knowledge were not analysed or assessed. In both cases, the knowledge content that was used was not described or could not be determined. Hence, managing knowledge in innovation processes in these cases was ad hoc or not even present. And we argue that for reasons of not having the relevant knowledge and not giving the involved individuals proper steering and managing innovation is not possible. The two case studies show how difficult it is to manage the dynamics of knowledge in innovation processes if you do not use adequate levels of analysis.

Section 2.3 provides a literature review to embed the findings of the two case studies. We want to show that the situation in the case studies is consistent with the literature.

2.3 Literature review of Innovation, Project, and Knowledge Management

In the review, we analyse current methods and tools in innovation, project, and knowledge management literature with respect to knowledge in innovation processes. In the introduction section, we stated that individuals should be the ontologically lowest units of analysis to manage innovation. Literature on knowledge management shows that in theory individuals create knowledge in interaction with others.

The review is organized as follows. First, we discuss (a) existing perspectives on innovation management, (b) a definition of innovation, and (c) various phases of innovation processes. Second, we review the project management literature to understand how current tools support management to plan and follow innovation processes. Finally, we review the knowledge management literature to find out whether its current approaches can be applied in the innovation context.

2.3.1 Managing the innovation project

In innovation management literature three main streams with regard to the determining factors of innovation exist: (1) an individualist perspective, (2) a structuralist perspective, and (3) an interactive process perspective.

In the individualist perspective (Slappendel 1996), personality traits of innovating individuals are determinants for innovativeness. In the structuralist perspective, the organization and its relation to the environment determine innovativeness. The interactive process perspective uses the individual and his activities, but also the organizational structure. Research on innovation from the interactive process perspective "involves the description and analysis of temporal sequences of activities which occur in the development and implementation of innovations" (Pierce *et al.* 1977; Garcia and Calantone 2002). The interactive process perspective takes into account the individual, his tasks, and his environment, i.e. the organizational structure. However, it does not acknowledge the influence of knowledge, and the dynamics of knowledge that is normal within innovation processes. The interactive perspective is also a good starting point for the complementary conceptual framework we present in section 4.

Innovation defined

Change, innovation, invention, creative behaviour, and adaptation have often gone undefined. On other occasions, they have been interchangeably used. Our definition of innovation is a combination of innovation definitions by Pierce and Delbecq (1977), Zaltman, Duncan, and Holbek (1973), Rogers (1995), and West and Farr (1990). We see innovation as "a complex multi-phased activity, where an artefact moves from initiation to adoption and implementation within a unit of adoption." This artefact – a product, process, idea, service, architecture, practice, or material artefact – is new to the unit of adoption or to the innovating actors and is designed to significantly benefit the unit of adoption or a possibly larger context.

Innovation process and phases

Most innovation processes are divided into phases. Phasing is important because it provides management with anchors to assess the progress of the innovation process. In innovation management literature, different authors use different phases of the innovation process. For example, Haner (2002) divides the innovation process into idea generation, screening, evaluation, and implementation. Other authors use similar phases. In these phases, similarities as well as differences can be found. For instance, the generality of the phases in the innovation processes differs. Table 1 provides an overview of the various process layouts. The precise division into phases in the innovation process may vary as long as it is suitable to the specific environment or industry. In many cases innovation processes are organized as projects. It is, therefore, not surprising that similar sequences of phases can also be found in the project management literature. In projects, phasing often is directed towards a specific environment or industry, for instance the design of information systems.

Management tasks

According to Van de Ven (1999; see also: Garcia and Calantone 2002; Dvir, Raz, and Shenhar 2003), management of innovation processes should vary its role, activities, and involvement in the different phases of the innovation process. However, innovation management literature does not specify the management tasks at a level that can be directly applied to the work floor level, the individuals, and the tasks.

It is generally believed that the project management task is based on the assumption that performance or end product goals are always clear and well defined in advance (Dvir, Raz, and Shenhar 2003). In this view, all the project manager has to do is prepare a solid project plan and follow this plan all the way to success. In an empirical study on current practices in project management, White and Fortune (2002; see also Steyn 2002; Herroelen and Leus 2001) found that most project managers use in-house project management methods and "projects in controlled environments" (PRINCE). Furthermore, they report Gantt bar charts, work breakdown structures (WBSs), and critical path methods as the most frequently used project management tools.

Plan the organization

Project management uses tools to plan the organization of processes. In most projects, this plan is divided into an organization plan – which consists of a WBS and an organization breakdown structure (OBS) – and a project schedule.

The WBS defines the tasks that have to be carried out to realize the project aim. In the WBS for each task, aims, i.e. task conditions a posteriori (afterwards), as well as interdependencies with other tasks, are defined. The OBS allocates roles, or, if possible, persons, to the tasks that are defined in the WBS.

Table 1. Innovation processes

(Chiesa, Coughlan, and Voss 1996)	(Pahl and Beitz 1996)	(Buijs 1987)	(Frame 1995)	(Krishnan and Ulrich 2001)	
(1) Idea generation	(1) Concept generation	(1) Prepare project	(1) Determine path	(1) initiation	(1) Product strategy and planning
(2) Screening	(2) Product develop-ment	(2) Analyse actual state	(2) Determine goals	(2) defini-tion	(2) Product development organization
(3) Evaluation	(3) Produc-tion process innovation	(3) Deter-mine target state	(3) Develop-ment,	(3) realiza-tion	(3) Project management
(4) Implemen-tation	(4) Tech-nology ac-quisition	(4) Develop solution ideas,	(4) Implemen-tation result-ing into search fields, design goals, product designs, alter-native product and market strategies respectively		(4) Concept development
	(5) leader-ship process	(5) Deter-mine solu-tions			(5) Supply chain design
	(6) resource provision	(6) Realize solutions			(6) Product design
	(7) system and tools provision				(7) Perform-ance testing and validation
					(8) Produc-tion ramp-up and launch

Two approaches exist to schedule projects. In the first approach, the task sequence with the longest duration determines the duration of the entire project, for example the "critical path method" and "the critical chain and buffer management" (CC/BM) algorithms (Soroush 1994). The second approach identifies the success probability of the various task sequences in the innovation project, given the predetermined start and due date (deadline); for example the "most critical path" algorithm (e.g. Schreiber *et al.* 2002).

Current project organization plans cannot be used to manage knowledge in an innovation context. Knowledge is needed to carry out a task (Turner 1987). However, the WBS lacks the attribution of knowledge needs to a task. Furthermore, the (sub)tasks knowledge creation and knowledge sharing are not explicitly included in the WBS.

The organization plan, that is the WBS and OBS together, cannot be based on a match of tasks' knowledge needs and persons' knowledge repository in an innovation process. In reviewing the project management literature, we found no explicit method to allocate persons to tasks based on someone's knowledge or skills.

Optimization of the organization plan – taking into account the knowledge creation and knowledge sharing tasks and capabilities of the involved individuals – is only possible if management plans the WBS and OBS iteratively. Current project management tools and methods lack this approach. Optimization of the project schedule exists in case of an optimal project organization plan: a plan that takes into account these knowledge-related aspects.

Sometimes it is necessary in innovation processes to first create or share knowledge to be able to carry out a certain task. The duration of these tasks depends on the allocation of a person to this task. For example, it is likely that the duration of the knowledge creation task is shorter if an expert in a relevant field carries out the task compared to a novice in this field. The knowledge sharing between two tasks also relies on the WBS and OBS. For instance, knowledge sharing between two tasks is easy if the same person carries out both tasks.

Follow the project

With the help of project management tools, management wants to follow a project. That is, management monitors, stimulates, and facilitates a project after project initiation. However, what management should be able to monitor and intervene in are activities at the individual and task level in order to follow, stimulate, and facilitate the persons who are involved in the project. However, management monitors (or assesses) ex post and ex ante criteria of the project phase that is terminated and of the project phase that will start, respectively. Based on this assessment, management allows or does not allow the start of the next phase, that is, to carry out the next cluster of tasks.

Projects can be phased as a waterfall, for instance as in SDM (Kusiak and Wang 1993), but they can also be phased parallel, for instance in concurrent engineering (Cooper 1994), or they can be phased with fluid gates

(Alvesson 2001). Project management methods, such as PRINCE2, often predefine the project phases. However, PRINCE2 does not take phasing – the definition of task clusters – of knowledge (dependencies) into account.

In the innovation process, often the creation and availability of knowledge are ex post as well as ex ante conditions for project tasks. The current project management literature does not discuss a method or tool that can be used to assess these knowledge-related conditions. For instance, CC/BM improved "classical" WBS/OBS planning by including resource dependencies, but CC/BM does not focus on knowledge.

Current project management tools and methods monitor whether the project team met the ex post criteria in a certain project stage. Therefore, interventions are directed at the team level, which then have to be translated into persons that carry out the knowledge tasks of the team. Thus, interventions only indirectly influence the individual that carried out a bottleneck task. If the project management were able to monitor whether an individual met the ex post criteria of a knowledge task, management interventions could be more direct and hence more focused.

2.3.2 Knowledge management

In earlier sections, we discussed knowledge management and the creation, utilization, and sharing of knowledge from the perspective of innovation and project management literature. This section discusses managing knowledge in innovation processes from the perspective of knowledge management (literature) itself.

Knowledge management can be seen as a container term for a wide spectrum of academic orientations (Alvesson 2001). Authors struggling with the concept [knowledge management] typically slide either to a "knowledge" or to a "management" pole, or move away from what may be seen as the usual meanings of these two labels (Alvesson 2001). The classical formulation of management by Fayol (1987; see also: Taylor 1997; Beesley 2004; Birkinshaw, Nobel, and Ridderstrale 2002; Schreiber *et al.* 2002; Teece 2000) is as follows:

> The managerial function seeks to derive optimum advantage from all available resources and to assure the smooth working of the six essential functions including the managerial function itself (p.13). To manage is to plan, organize, coordinate, command, and control (p.13). Management … is an activity spread across all members of the "body corporate" – the total personnel structure of the organization.

Following Fayol, we view knowledge management as the planning, organizing, coordinating, commanding, and controlling of knowledge – or the individuals as bearers of knowledge who utilize, create, or share knowledge to carry out their tasks – to derive optimum advantage from all available resources and to assure the smooth working of all activities to which organizational activities give rise. Knowledge management is an activity spread across all members of the organization.

Knowledge classifications

Earlier (section 3.1.2), we indicated that the identification of proper knowledge domains beforehand (ex ante) and the knowledge types afterwards (ex post) is difficult. In the literature, several classifications of knowledge content and knowledge types are used.

Knowledge content often is referred to as assets (Szulanski and Amin 2001), disciplines (Chen and Paul 2001), domains (Hall and Andriani 2003), skills (Von Krogh and Roos 1994; Hellstrom 2000), and competencies (Boisot 1995). Skills and competencies are often related to the use of knowledge of a certain content.

The knowledge management literature displays a wide variety of classifications of knowledge forms or types, including coded and uncoded, abstract and concrete, diffused and non-diffused (Pylyshyn 1984), declarative and procedural (Polanyi 1967; Nonaka 1994), and tacit and explicit (2003) knowledge. We refer to Cijsouw and Jorna (2003) for a discussion of the differences and similarities of these knowledge-type classifications. Up to now, knowledge management literature lacks a golden standard for the classification of knowledge content and of knowledge type or form. The classifications are also ambiguously used in the literature. We come back to this discussion in section 4 in which we present a division in sensory, coded, and theoretical knowledge.

Knowledge tasks of individuals in innovation projects

In innovation projects individuals do not "possess" all the knowledge they have to utilize beforehand. Knowledge acquisition is needed. An individual can acquire knowledge by means of knowledge creation himself or by knowledge sharing. Thus, an individual has three different knowledge tasks: creation, sharing, and utilization. These three different knowledge tasks are connected to the three main streams that focus on the realization and improvement of: (1) knowledge creation (e.g. Argote and Ingram 2000; Szulanski 2000; Hoopes and Postrel 1999), (2) knowledge sharing (e.g. Taylor and Lowe 1997; Teece 2000), and (3) knowledge utilization (e.g. Alavi and Leidner 2001; Rubenstein-Montano *et al.* 2001). We argue that in innovation processes, management of knowledge should realize and improve knowledge creation, knowledge sharing, as well as knowledge utilization.

The use of information and communication technology (ICT) to store knowledge is frequently addressed in the knowledge management literature (Roth 2003). For this reason, knowledge storage often is the centre of a fourth knowledge management stream. This stream incorrectly equals knowledge management to information management (Jorna 1998). However, knowledge storage most often belongs to the knowledge sharing stream. If knowledge sharing is ICT supported, for instance using Intranet as a medium, then knowledge storage is a natural part of the knowledge sharing task.

However, it is interesting to see here again that in the three knowledge management streams, the team or the organization, and not the individual is the unit of analysis. We will first discuss details of knowledge management in the three streams and then establish some omissions and deficiencies and relate these to the case studies. In the discussion we follow the "natural" order of knowledge creation, sharing and utilization.

Knowledge creation

Knowledge creation is often discussed at the level of the organization (Von Krogh, Ichijo, and Nonaka 2000) or the team (Nonaka 1994). Authors that take the individual into account adopt an interactionist approach: the individual creates knowledge in interaction with its environment. This emphasis on the team or organization level is strange, because knowledge creation and creativity are closely related. As a cognitive psychologist, Boden (1994) distinguishes historical creativity (H-creativity) and psychological creativity (P-creativity). H-creativity applies to ideas that are fundamentally novel with respect to the whole of human history (Boden 1994). P-creativity concerns ideas that are fundamentally novel with respect to the individual mind that had the idea (Csikszentmihalyi 1999). P-creativity is the type of creativity that is used in most innovation processes.

Creativity occurs when a person makes a change in a domain, a change that will be continued through time (Csikszentmihalyi 1999). Changes are not adopted, unless they are sanctioned by a group entitled to make decisions as to what should or should not be included in the domain (Nonaka 1994). The individual creates an idea, i.e. a possible change in a domain, and then a group or groups within society justify it.

The interaction between the individual and society – his environment – can be a catalyst for knowledge creation as well: "Organizational knowledge creation … should be understood in terms of a process that 'organizationally' amplifies the knowledge created by individuals, and crystallizes it as a part of the knowledge network of organizations" (Nonaka, Toyama, and

Noboru 2000; Nonaka and Toyama 2003). Nonaka (1994) conceptualizes organizational knowledge creation in an interaction space (ba) in which individuals create knowledge, converging knowledge types through socialization, externalization, combination, and integration. This interaction space consists of a physical, temporal, and context dimension; people create knowledge in a conceptual space-time-context continuum. Thus, knowledge creation is an activity of the individual in interaction with others. It is revealing that creativity is not explicitly valued in innovation management methods.

Knowledge sharing

An individual – actor – can acquire the knowledge to carry out his task through knowledge sharing and through knowledge transfer. Often, the terms knowledge sharing and knowledge transfer are used interchangeably. However, the difference between knowledge sharing and knowledge transfer is the direction of the knowledge exchange. In knowledge sharing, knowledge is exchanged in both directions, whereas in knowledge transfer, knowledge is exchanged in one direction, from the "knowing person" to the "not (yet) knowing person". This section is limited to discussing knowledge sharing, because knowledge transfer is a subset of knowledge sharing. Discussing knowledge sharing already involves many aspects of knowledge transfer.

In knowledge management literature (Carlile 2002), knowledge sharing is often discussed at the level of the team or organization. Knowledge sharing is rarely discussed in direct relation to individuals who utilize this knowledge in order to fulfil tasks. Most articles relate knowledge sharing to communities of practice, intrinsic motivation, trust, etc., and not to the operational level, i.e. a task that has to be carried out. Carlile (2002) forms an exception to this tradition; he describes a model of knowledge transfer from one actor to another related to the task that has to be carried out.

Several scholars used Shannon and Weaver's (1963) communication model (see Cijsouw and Jorna 2003) to model knowledge sharing. In these models actor I – the "knowing person" – sends a "knowledge package" from its own knowledge repertoire through a medium to actor II – the not yet "knowing person" – who receives and interprets the "knowledge package" and incorporates this "new knowledge" into his own knowledge repertoire. Perhaps actor II utilizes this knowledge to carry out a task. The incorporation of knowledge to an actor's knowledge repertoire is a form of – as Boden called it – P-creativity.

Using this knowledge sharing model, five possible thresholds can be identified. First, actor I may send the "wrong" knowledge. Second, actor I may not send any knowledge at all. Third, the medium – often related to the knowledge type as we argue in section 4 – may be insufficient, that is, the "knowledge package" does not arrive at actor II. Fourth, actor II may not recognize the "knowledge package" and ignore it. Fifth, actor II may interpret the "knowledge package" incorrectly. It is possible that in case of knowledge sharing some feedback mechanism may inform the actors on the success of the knowledge exchange. In case of (the one-directional) knowledge transfer, no possibility for feedback exists. Hence, knowledge transfer has in general a higher risk of failure than knowledge sharing. If knowledge, individuals and tasks are not the primary focus of attention, management often misses these thresholds.

Knowledge utilization

We argue that knowledge utilization should be the central task for all individuals, because only this knowledge task directly contributes to the realization of innovations. For this reason, it is surprising that the knowledge management literature seems to accept the utilization of knowledge as beyond its scope. How knowledge utilization occurs is not discussed. The mechanisms of utilizing knowledge into action have been studied in psychology, for instance with respect to decision-making, or problem solving. It assumes that persons perceive something and act based on this perception; a perception-action link exists. In a decision-making perspective, all behaviour/activities are results from a person's decisions. Unfortunately, in knowledge management and innovation literature here the further elaboration or operationalization of knowledge utilization stops.

2.3.3 Conclusion: the overlap between the two case studies and literature

In the two case studies, we showed the problems with regard to the management of knowledge in innovation processes. Neither individuals, nor the perspective of tasks are taken into account. The interpretations of the cases are in line with the omissions in the innovation, project, and knowledge management literature.

In the innovation management literature, we found the following. First, innovation management lacks a special attention for knowledge. Second, the individual often is not the unit of analysis with the exception of roles, personality traits and related activities. Knowledge creation, utilization, and sharing are not included as one of these activities. Three, innovation management lacks consent on the layout and phasing of innovation processes.

In project management, i.e. the management at the work floor level of the innovation process, we found that (a) WBS lacks an attribution of knowledge needs to tasks. For this reason, (b) tasks cannot be decomposed into knowledge tasks: knowledge creation and knowledge sharing are not explicitly included in the WBS plan. (c) The OBS cannot allocate roles, or persons, to tasks based on a match between their knowledge and the tasks' knowledge needs. This is due to the lacking attribution of knowledge needs to tasks in the WBS. From a knowledge perspective, (d) the WBS and OBS cannot be optimized if they are not planned iteratively; only then a person's knowledge and the knowledge need of the task can be taken into account.

In our evaluation of knowledge management (see section 3.2.), we found five reasons why current knowledge management methods cannot be directly applied to the management of innovation processes. First, too much variety in classifications of knowledge content and knowledge types exist. Second, the unit of analysis is either the team or the organization, not the individual. Three, knowledge management itself is ill-defined. Four, current knowledge engineering methods treat knowledge as static. Five, knowledge creation is studied from an interactionist approach at the organizational level that does not adequately take into account how individuals create knowledge. Furthermore, we identified five thresholds in knowledge sharing based on the Shannon and Weaver communication model.

From the various reviews, we can infer common causes why current knowledge management, innovation management, and project management methods and tools are difficult to use to manage knowledge in innovation processes. In combination with the analyses of the case studies, we reformulate the causes as five omissions. They are:

1. The individual who carries out a task is not the unit of analysis. The organization and the team are the units of analysis.
2. Project management methods do not look at knowledge. For this reason, WBS and OBS plans are not based on all relevant criteria.
3. No method or tool supports management to obtain insight in the knowledge repertoire of the involved individuals.
4. Knowledge creation is studied from an interactionist approach at the organizational level. This does not take into account the dynamics of individuals who create knowledge and the dynamics of knowledge itself.
5. The variety of classifications of knowledge content and especially knowledge types/forms are not suitable to model knowledge creation.

2.4 The Semio-Cognitive Framework

We believe that the absence of an individual, knowledge, and task orientation is systematic in the management of innovation projects. We also believe that organizational semiotics can overcome these deficiencies for the following three reasons. First, information and knowledge exchange between individuals is in terms of signs and symbols. Second, the knowledge itself that is created individually or collectively in the early phases of the innovation requires the production of signs, whether it concerns words, pictures and sketches, or mathematical symbols. Third, an organization whether it concerns an R&D unit or a small innovative group, is itself the cause for as well as the consequence of the production and exchange of signs. Organizational semiotics as the study of signs and sign understanding in an organizational context can deal with knowledge creation and knowledge production by individuals in innovation projects. We will illustrate the relevance of organizational semiotics by describing a semiotic framework that we develop for the study of innovation projects.

The framework starts with and uses the individual as the unit of analysis. The framework derives ideas and concepts from the knowledge and business process models in CommonKADS (Schreiber *et al.* 2002) and from the metaphor of the information space (Boisot 1995). The cognitively oriented framework is intended to work as an additional tool for the management of knowledge in innovation processes besides the current innovation, project, and knowledge management methods. This (semio)-cognitive framework looks at individuals as human information processing systems. Humans as cognitive systems create, share, and use all kinds of knowledge.

The framework is an information processing and task model that provides the possibility to plan and follow the knowledge that individuals use to carry out their tasks. In this framework, we can also use the classifications of knowledge content and knowledge types. We believe that these classifications are best suited to grasp the dynamics of knowledge at the individual and inter-individual level.

2.4.1 The business process model

In the framework, business processes consist of tasks. These tasks consist of one or more knowledge oriented tasks: knowledge creation, sharing, and utilization. Only individuals create and utilize knowledge. This is normally done in interaction with others. Therefore, the unit of analysis is the individual and his relations. Knowledge sharing involves at least two individuals. Hence, the relation between the involved persons is important.

The framework makes it possible to identify the knowledge sharing thresholds (see section 3.3.2.3), because the knowledge repertoires of individuals as well as the relations between individuals are included.

To carry out a knowledge task, an individual uses his cognition, his knowledge repertoire. A knowledge repertoire consists of content knowledge that is presented in a certain form or type.

2.4.2 Knowledge content classification

The framework classifies knowledge content in domains. A knowledge domain comes closest to a single interconnected cluster of knowledge. Fields of science are good examples of knowledge domains, e.g. medical sciences, economy, or sociology. A knowledge domain may consist of (a combination of) "skills", "procedures", "facts", etc. In the framework, we do not need this level of detail for the denomination within the knowledge domain.

Within a knowledge domain, other interconnected clusters may exist. For instance, in the medical sciences, several specializations exist, such as ENT (ear, nose, throat), thorax surgery, or orthopedics. If the medical sciences were the starting point, then it would be possible to model the specializations as subdomains. If the specialization was the starting point, then the specializations were the domain and the medical sciences could be modelled as a metadomain. However, we prefer to avoid semantic confusion. For this reason, we only refer to domains. It is possible to adjust the meaning of a "domain" in the knowledge framework during an innovation project without a need for relabeling.

2.4.3 Knowledge-type classification

The framework classifies the knowledge types (or presentations) along three non-orthogonal axes that form a knowledge space: sensory (ranging from rough to detailed), coded (ranging from weak to strong), and theoretical (ranging from concrete to abstract) knowledge. The framework is semiotic, because all knowledge is expressed in signs and symbols, from indexes, icons, characters to diagrams and notations. In our types of knowledge we elaborated upon concepts developed by Boisot (1995, 1998), who uses "codedness" and "abstraction" in his information space. However, Boisot adds a "diffusion" dimension to form a three dimensional space. Because the diffusion dimension is beyond the individual level, we leave it out of the framework.

Sensory knowledge forms the first dimension in the knowledge space. Sensory knowledge is the knowledge a person obtains using sensory organs. The knowledge is as concrete as the event that is interpreted. It is behaviour. Examples of such knowledge are the smell of spices, or the sound of a bird's whistle, or the knowledge of somebody's face. The first dimension ranges from rough to detailed sensory knowledge. In detailed sensory more fine-grained and specific sensory aspects are present. Sensory knowledge expresses itself often in skills or procedures of behaviour.

Coded knowledge is the second dimension in the knowledge space. Coded knowledge is the group or category that is formed on top of the knowledge of a concrete event – the sensory knowledge. Coded knowledge means using signs; the concrete event becomes a sign. Words, diagrams, and pictograms are all examples of such codes. Coded knowledge can be used apart from the concrete event it refers to; it allows the description of a smell of the spice without the presence of this smell. Coded knowledge forms a dimension that ranges from weak (picture) to strong (math). The dimension from weak to strong is indicated by a decreasing ambiguity; the stronger the code the less ambiguous the transferred knowledge is.

Theoretical knowledge is the structure that can be formed on top of sensory and coded knowledge. All knowledge that reflects a structure, method, or pattern is theoretical. For example, physical laws are theoretical knowledge, but ideological or religious coherent structures are theoretical knowledge as well. Theoretical knowledge can be made visible in asking and answering "why" questions. This third dimension in the knowledge space ranges from concrete to abstract theoretical knowledge; concrete theoretical knowledge consists of small "why-chains", whereas abstract theoretical knowledge consists of long and complex chains.

Figures 1 and 2 depict examples of knowledge spaces. Figure 1 is a static example. In this knowledge space – a snapshot of one moment in

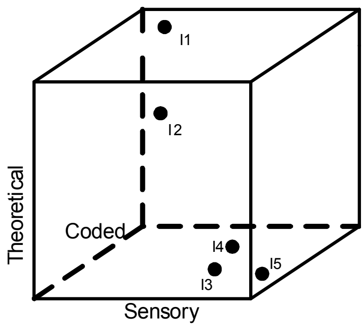

Fig. 1. Depiction of five individuals (I_1 to I_5) on knowledge domain A at one moment in time

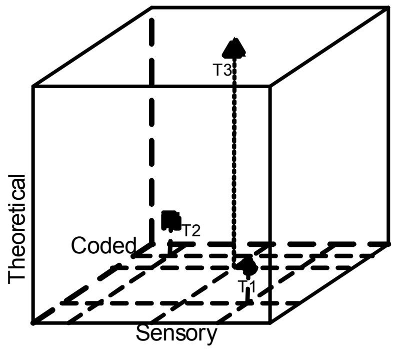

Fig. 2. One individual's knowledge types on knowledge domain a moving through time

time – the individuals (I_1–I_5) who are involved in the process at that moment are situated according to the knowledge types they use to carry out their tasks. One knowledge domain A is presumed in Figure 1. Figure 2 depicts the conversion through time of the knowledge that one individual I_1 uses. The bullets in the space refer to T_1 to T_3. This conversion is based on learning or development of I_1. First, the individual uses mainly detailed sensory knowledge (circle; right on 1st axis), then shifts to the use of mainly strongly coded knowledge (square; behind on the 2nd axis), and at the third moment (triangle; on top of the 3rd axis), strongly coded knowledge is used in combination with abstract theoretical knowledge. Note that theoretical knowledge is not used before coded knowledge has been acquired and that coded knowledge builds upon sensory knowledge in this conversion of knowledge.

2.4.4 Example

The following example is intended to provide a better understanding of the way the knowledge classification in sensory, coded, and theoretical types is tightly connected to the creation, sharing, and utilization of knowledge at the individual level. The example follows a natural path of knowledge creation to show that the proposed knowledge-type classification is a contribution to knowledge management theory.

If a song is often on the radio (the concrete event), people start to recognize it and eventually distinguish more and more details (utilization of sensory knowledge). If you want to tell persons which tune you are so excited about, they have to be present during the song; otherwise they will not be able to hear it. Having heard the song many times you might be able to sing it without the stimulus of the song on the radio (another utilization of sensory knowledge). Then everybody recognizes the song; at least that is, if you sing it accurately and in tune – if you add enough details.

Learning that this song is entitled “Let it be” and was originally performed by the Beatles provides a label to the tune. Now you can tell other persons that “Let it be” was on the radio again. Many persons will know what you have heard. They know this song as “Let it be” by the Beatles – in this case coded knowledge makes it possible to discuss the event – talk about it in terms of language codes – without the need of presence if the song is on the radio, or without having to listen someone’s singing qualities.

The musical score of “Let it be” contains the codes of the song in more detail: lyrics, instrumentation, melody, and chord progressions. This is only useful if you are able to read musical scores well. If this is the case, you can sit down with the “Let it be” score, start reading and then the song fills your head. If you use musical scores, it is possible to play this song with other persons; some of them may never even have heard the song, but because the musical score contains rather unambiguous codes – i.e. strong codes, or notations – they will play it correctly.

A song consists of various codes: lyrics, melody, chords, chord progressions, and instrumentation. These codes are structured in a certain way; hence, the song contains “verses”, “choruses”, and “bridges”. This is what we call theoretical knowledge. These structures can be used to analyse the Beatles repertoire answering such as “*why* is the Beatles repertoire popular”. Software such as *hit song science* claims to be able to answer this question using a mathematical pattern in melody, tempo, rhythm, pitch, and chord progression; if a song lands in one of the four “hit clusters” it has hit potential.

2.5 Discussion: The Framework and the Omissions

This chapter introduces a framework that intends to better support the management of knowledge in innovation processes than the current methods and tools. In the cases described in section 2 and in the current literature on innovation, project, and knowledge management (section 3), we identified five omissions regarding the management of knowledge in innovation processes. This section benchmarks our framework with the current methods using the five omissions as guidelines.

Omission I: The individual who carries out the task is not the unit of analysis. The organization and the team are the units of analysis. The case studies and the literature review illustrated that the individual should be the unit of analysis to be able to manage the dynamics of knowledge, that is knowledge creation and transfer in innovation projects.

Therefore, the framework takes the individual with his tasks and knowledge as the ontologically lowest unit of analysis. Thus, the framework eliminates this omission in the current literature. Of course, from the individual level the cognitive framework can aggregate to ontologically higher levels such as the team or the organization. In this case, it is important that the framework also includes the interaction-relations of the individuals involved in the innovation process.

Omission II: Project management methods do not consider knowledge. Therefore, the WBS and OBS plan are not based on all relevant criteria. The fact that WBS and OBS do not include knowledge, results in imprecise estimations of tasks, and duration, among others. Not only does the cognitive framework include knowledge, it can also be used to plan WBS and OBS. This means that the framework includes individuals that participate in the process – relevant to OBS – and tasks that are carried out in the innovation process – relevant to WBS. Furthermore, it includes the individuals' knowledge repertoires, which are used to optimize the WBS and OBS plan.

The match between the knowledge needs and repertoires determines the need of knowledge sharing and knowledge creation in the innovation process. If the match is optimal, these tasks do not have to be carried out. The current project management tools and methods do not decompose tasks into these knowledge tasks because they lack a knowledge view. We believe that in the practice of innovation projects the unforeseen addition of knowledge creation and knowledge sharing to knowledge utilization lengthens the project duration: more unforeseen (sub)tasks have to be carried out in order to realize the innovation.

Omission III: No method or tool supports the management to obtain insight in the knowledge repertoire of the involved individuals: Innovation processes are divided into phases to give the management anchors for monitoring the progress. In fact, monitoring is an assessment whether the ex post criteria of the former phase or the ex ante criteria of the next phase are present. This assessment determines whether it is allowed to start the next phase. Knowledge is not a criterion in the assessments in the current project and innovation management methods.

The framework eliminates this omission because it identifies the knowledge that an individual needs to carry out each task. The presence of this knowledge – which can be specified in content and type – in someone's knowledge repertoire can be added to the task criteria. The framework supports the assessment whether a person that is intended to carry out a certain task has knowledge of the right content and type in his repertoire. If this is not the case, the management is able to intervene directly at the task and individual level. Furthermore, the cognitive framework can support the persons who carry out a task to assess whether the knowledge-related ex ante and ex post conditions are fulfilled. In this situation, knowledge management becomes an activity that is spread across all members of the organization.

Omission IV: Knowledge creation is studied from an interactionist approach at the organizational level. This does not take into account how individuals create knowledge: The current literature on knowledge creation takes an interactionist perspective. It leaves aside that as a start knowledge creation takes place at the individual level, in fact, inside the individual. This will often be in interaction with others. Our framework also eliminates this omission. The framework has the individual as the ontologically lowest unit of analysis. It takes into account the dynamics of knowledge creation at the individual (and task) level and – very important – also over time. The framework allows the observation of the changing knowledge content and converging knowledge types due to knowledge creation. Furthermore, the framework models the interaction-relations of individuals. For this reason, our framework can be used within the interactionist approach.

To create knowledge in a new domain, an individual utilizes knowledge of closely related domains. The knowledge formed in a newly created domain may change from the sensory to the coded and from there to the theoretical dimension. However, theoretical knowledge has to be created upon coded knowledge that on its turn has to be created upon sensory knowledge in the domain (the content) in which knowledge is created. The "cognitive" distance between the knowledge a person already has and the knowledge that has to be created, indicates the difficulty of the knowledge creation task; this determines the duration of this subtask to a large extent.

The framework acknowledges that individuals create knowledge in interaction with others. It models the interaction-relations of individuals. Our cognitive framework can be used to get insight, or even to compare, the knowledge repertoires of the individuals in these interaction-relations. Knowledge gaps can be identified at the (ontological) level of individuals, teams, or organizations. Such gaps are potential problems in the knowledge creation interaction-relation. The management may choose to add these knowledge creation relations to their monitoring tasks and perhaps prevent failing knowledge creation interactions.

Omission V: The variety of classifications of knowledge content and especially knowledge types/forms is not suitable to model knowledge creation: The knowledge management literature contains many classifications of knowledge contents and types. The classifications that are used in case of media/bearers of knowledge content – skills, competencies, assets, and domains – do not exclude each other. The framework classifies knowledge content into knowledge domains. The interconnectedness of the knowledge content cluster is the only demand for knowledge domains. How these domains relate to each other is important to determine the "cognitive" distance. We did not find an objective and reproducible method to determine the distance between knowledge domains. Hence, the management can only estimate this distance based on their gut feeling and past experiences and in the sense that they are experts in the domain.

The classifications of knowledge types often are dichotomies, or extremes at a dimension. In these cases, a knowledge type shows the tasks that the knowledge is utilized. These classifications are difficult to use during the creation of knowledge.

Our framework classifies knowledge types into sensory, coded, and theoretical knowledge. This classification is derived, among others, from cognitive psychology and seems to be a closer fit to the actual creation of knowledge than the classifications that were found in the knowledge management literature. Thus, the framework improves the management's possibility to model knowledge creation with respect to both knowledge content and appearance.

In conclusion, the cognitive framework provides the management, as well as the individuals involved, an innovation process insight into knowledge repertoires. Insight in knowledge repertoires also includes which knowledge has to be created and transferred by whom to be able to carry out certain tasks. Knowing knowledge repertoires of individuals and the knowledge content and type they have to create is an input to WBS and OBS design that is currently lacking. We believe this provides the management with good indicators for the duration of knowledge creation and utilization tasks. With respect to the knowledge transfer task, it is now also easier to identify the five possible thresholds using the cognitive framework: (1) send the "wrong" knowledge, (2) not send any knowledge at all, (3) the medium is insufficient, (4) the "knowledge package" is not recognized and hence ignored, and (5) the "knowledge package" is interpreted incorrectly.

The proposed cognitive framework makes it easier to monitor the innovation process, because it can be observed who carries out knowledge tasks at the level of the individual. It emphasizes the individuals' abilities to create, utilize, and share knowledge in the team composition. This may counterbalance the current emphasis in the project management literature on personality types, which is about roles and not about knowledge.

Currently in innovation management and project management, often a top-down management approach is applied; the management designs the project and the employees (researchers) have to carry out their tasks according to the design. In the above, we have indicated how our framework can be used in addition mainly to these top-down approaches. However, the cognitive framework also gives the opportunity to introduce a bottom-up approach in the management of innovation processes for two reasons. First, the individual is better capable of assessing his knowledge repertoire than anybody else. Hitherto, it is impossible to identify the knowledge content or types one has without the cooperation of this person.

Secondly, the framework may prove to be valuable in a network perspective on innovation management, because it gives the opportunity to include the content of the nodes (the individuals) in the network. In using the framework, the content of a node is modelled as a detailed knowledge (mind) map of an individual, team, or organization. A combination of a network approach and the cognitive framework might prove an interesting direction for future research.

Although the two conceptual directions of bottom-up and networks we identified are relevant, we believe the research on knowledge in innovation really also lacks empirical data at the level of the individual and the task. Using the cognitive framework, we have carried out longitudinal empirical studies for a 2-year period in four medical device development projects. These projects involve individuals from various disciplines employed by various organizations. The first project is in the design and prototyping phase. It aims at developing a voice prosthesis that should support laryngectomized patients. In the second project an intravascular oxygenator is developed. This project is currently in the transfer phase from university to industry. The third project aims at the introduction of an organ perfusion system to the market. It is also being transferred from university to industry. In the fourth project, a heart-assist device is designed. A start-up company carries out all the development of the device, the animal studies, as well as the clinical studies. The results of these empirical projects (also see Cijsouw 2006) strengthen our belief in the cognitive framework for innovation and knowledge management. Knowledge must be assessed and not just as another kind of resource. It is the input, throughput and output of innovation, it is dynamic and it is cognitively bounded.

References

Alavi M, Leidner DE (2001) Review: knowledge management and knowledge management systems: conceptual foundations and research issues. MIS Quarterly, 25, pp. 107–125

Alvesson M (2001) Odd couple: making sense of the curious concept of knowledge management. The Journal of Management Studies, 38, p. 995

Argote L, Ingram P (2000) Knowledge transfer: a basis for competitive advantage in firms. Organizational behavior and human decision processes, 82, pp. 150–169

Barras R (1990) Interactive innovation in financial and business services: the vanguard of the service revolution. Research Policy, 19, pp. 215–237

Beesley L (2004) Multi-level complexity in the management of knowledge networks. Journal of Knowledge Management, 8, pp. 71–88

Birkinshaw J, Nobel R, Ridderstrale J (2002) Knowledge as a contingency variable: do the characteristics of knowledge predict organization structure. Organization Science, 13, pp. 274–289

Boden MA (1994) Précis of the creative mind: myths and mechanisms. Behavioral and Brain Sciences, 17, pp. 519–570

Boisot MH (1995) Information Space: A Framework for Learning in Organizations, Institutions and Culture, Routledge, London

Boisot MH (1998) Knowledge Assets: Securing Competitive Advantage in the Information Economy, Oxford University Press, Oxford

Brown L, Eisenhardt KM (1995) Product development: past research, present findings, and future directions. The Academy of Management Review, 20, p. 343

Buijs J (1987) Innovatie en Interventie, 2 edn, Kluwer, Deventer, The Netherlands

Carlile PR (2002) A pragmatic view of knowledge and boundaries: boundary objects in new product development. Organization Science, 13, pp. 442–455

Chen C, Paul RJ (2001) Visualizing a knowledge domain's intellectual structure. Computer, 34, p. 65

Chiesa V, Coughlan P, Voss CA (1996) Development of a technical innovation audit. Journal of Product Innovation Management, 13, pp. 105–136

Cijsouw R, Jorna R (2003) Measuring and mapping knowledge types. In: Gazendam HWM, Cijsouw R, Jorna R (eds.), Dynamics and Change in Organizations: Studies in Organizational Semiotics 3, Kluwer, Dordrecht, The Netherlands, pp. 215–243

Cijsouw R (2006) Knowledge in Medical Device Development Projects, PhD dissertation series of the SOM research school

Cooper RG (1994) Third-generation new product processes. Journal of Product Innovation Management, 11, pp. 3–14

Csikszentmihalyi M (1999) Implications of a systems perspective for the study of creativity. In: Sternberg RJ (ed.), The Handbook of Creativity. Cambridge University Press, Cambridge, pp. 313–335

Dvir D, Raz T, Shenhar A.J (2003) An empirical analysis of the relationship between project planning and project success. International Journal of Project Management, 21, pp. 89–95

Ernst D, Kim L (2002) Global production networks, knowledge diffusion, and local capability formation. Research Policy, 31, pp. 1417–1429

Evaristo JR, Scudder R, Desouza KC, Sato O (2004) A dimensional analysis of geographically distributed project teams: a case study. Journal of Engineering and Technology Management, 21, pp. 175–189

Fayol H (1987) General and Industrial Management, rev. edn, Pitman, London
Frame JD (1995) Managing Projects in Organizations. How to Make the Best Use of Time, Techniques and People, Jossey-Bass, San Francisco, CA
Frederiksen LF, Hemlin S, Husted K (2004) The role of knowledge management in R&D: a survey of Danish R&D leaders' perceptions and beliefs. International Journal of Technology Management, 28, pp. 820–839
Garcia R, Calantone R (2002) A critical look at technological innovation typology and innovativeness terminology: a literature review. Journal of Product Innovation Management, 19, pp. 110–132
Goldfarb B, Henrekson M (2003) Bottom-up versus top-down policies towards the commercialization of university intellectual property. Research Policy, 32, pp. 639–658
Hall R, Andriani P (2003) Managing knowledge associated with innovation. Journal of Business Research, 56, pp. 145–152
Haner UE (2002) Innovation quality – a conceptual framework. International Journal of Production Economics, 80, pp. 31–37
Hellstrom T (2000) Knowledge and competence management at Ericsson: decentralization and organizational fit. Journal of Knowledge Management, 4, p. 99
Herroelen W, Leus R (2001) On the merits and pitfalls of critical chain scheduling. Journal of Operations Management, 19, pp. 559–577
Hoopes DG, Postrel S (1999) Shared knowledge, "glitches", and product development performance. Strategic Management Journal, 20, pp. 837–865
Jorna RJ (1998) Knowledge management really is about representations. The Semiotic Review of Books, 9.2, pp. 5–8
Kazanjian RK, Drazin R, Glynn MA (2000) Creativity and technological learning: the roles of organization architecture and crisis in large-scale projects. Journal of Engineering and Technology Management, 17, pp. 273–298
Krishnan V, Ulrich KT (2001) Product development decisions: a review of the literature. Management Science, 47, pp. 1–21
Kusiak A, Wang J (1993) Decomposition in concurrent engineering. In: Kusiak A (ed.), Concurrent Engineering: Automation, Tools, and Techniques, Wiley, New York, pp. 481–507
Landry R, Amara N, Lamari M (2001) Utilization of social science research knowledge in Canada. Research Policy, 30, pp. 333–349
Meyer-Krahmer F, Schmoch U (1998) Science-based technologies: university-industry interactions in four fields. Research Policy, 27, pp. 835–851
Nonaka I (1994) A dynamic theory of organizational knowledge creation. Organization Science, 6, pp. 14–37
Nonaka I, Toyama R (2003) The knowledge-creating theory revisited: knowledge creation as a synthesizing process. Knowledge Management Research and Practice, 1, pp. 2–10
Nonaka I, Toyama R, Noboru K (2000) SECI, Ba and leadership: a unified model for dynamic knowledge creation. Long Range Planning, 33, pp. 5–34
Pahl G, Beitz W (1996) Engineering Design: A Systematic Approach, Springer, Berlin.

Pierce JL, Delbecq AL (1977) Organization structure, individual attitudes and innovation. Academy of Management Review, 2, pp. 27–37
Pisano GP (1996) Learning-before-doing in the development of new process technology. Research Policy, 25, pp. 1097–1119
Polanyi M (1967) The Tacit Dimension, Routledge & Kegan Paul, London
Pylyshyn ZW (1984) Computation and Cognition, MIT Press, Cambridge, MA
Rappert B, Webster A, Charles D (1999) Making sense of diversity and reluctance: academic-industrial relations and intellectual property. Research Policy, 28, pp. 873–890
Rogers EM (1995) Diffusion of Innovation, The Free Press, New York
Roth J (2003) Enabling knowledge creation: learning from an R&D organization. Journal of Knowledge Management, 7, pp. 32–48
Rubenstein-Montano B, Liebowitz J, Buchwalter J, McCaw D, Newman B, Rebeck K (2001) A systems thinking framework for knowledge management. Decision Support Systems, 31, pp. 5–16
Schreiber G, Akkermans H, Anjewierden A, de Hoog R, Shadbolt N, Van de Velde, W *et al.* (2002) Knowledge Engineering and Management: The CommonKADS Methodology, 3 edn, MIT Press, Cambridge, MA
Shannon E, Weaver W (1963) The Mathematical Theory of Communication, University of Illinois Press, Urbana, IL
Slappendel C (1996) Perspectives on innovation in organizations. Organization Studies, 17, pp. 107–129
Soroush HM (1994) The most critical path in a PERT network. Journal of the Operational Research Society, 45, pp. 287–300
Steyn H (2002) Project management applications of the theory of constraints beyond critical chain scheduling. International Journal of Project Management, 20, pp. 75–80
Swan J (2003) Knowledge management in action. In Holsapple CW (ed.), Handbook of Knowledge Management, Springer, Berlin, pp. 271–296
Szulanski G (2000) The process of knowledge transfer. Organizational Behavior and Human Decision Processes, 82, pp. 9–27
Szulanski G, Amin K (2001) Learning to make strategy: balancing discipline and imagination. Long Range Planning, 34, pp. 537–556
Taylor P, Lowe J (1997) Are functional assets or knowledge assets the basis of new product development performance? Technology Analysis and Strategic Management, 9, pp. 473–488
Teece DJ (2000) Strategies for managing knowledge assets: the role of firm structure and industrial context. Long Range Planning, 33, pp. 35–54
Tezuda S, Niwa K (2004) Knowledge sharing in inter-organisational intelligence: R&D-based venture alliance community cases in Japan. International Journal of Technology Management, 28, pp. 714–728
Tijssen RJW (2001) Global and domestic utilization of industrial relevant science: patent citation analysis of science-technology interactions and knowledge flows. Research Policy, 30, pp. 35–54

Turner WSI (1987) SDM: system development methodology, PANDATA, Rijswijk, Amsterdam

Van de Ven AH (1999) The Innovation Journey, Oxford University Press, New York

Von Krogh G, Roos J (1994) A perspective on knowledge, competence and strategy. Personnel Review, 26, p. 31

Von Krogh, G, Ichijo K, Nonaka I (2000) Enabling Knowledge Creation, Oxford University Press, Oxford

West MA, Farr JL (1990) Innovation and Creativity at Work: Psychological and Organizational Strategies, Wiley, Chichester

White D, Fortune J (2002) Current practice in project management – an empirical study. International Journal of Project Management, 20, pp. 1–11

Zaltman G, Duncan R, Holbek J (1973) Innovations and Organizations, Wiley, New York

Zellner C (2003) The economic effects of basic research: evidence for embodied knowledge transfer via scientists' migration. Research Policy, 32, pp. 1881–1895

Chapter 3
Viewpoint-Centred Methodology to Design Project/Subcontract Cooperation Policies

P.-J. Charrel[1,2], C. Thierry[1,3]

[1]*Université Toulouse 2 Le Mirail, Département de Mathématiques et Informatique, 5 Allée Antonio Machado, F-31058 Toulouse Cedex 9 France*
[2]*GRIMM-ISYCOM, Groupe de Recherche en Informatique et Mathématiques du Mirail/COMplex System Engineering Team*
[3]*ONERA-CERT, DCSD, 2 Av. Edouard Belin, BP 4025, F-31055 Toulouse Cedex 04 France*

Abstract

A methodology is presented to improve management projects which involve a project/subcontractor relationship and a shared resource. The project is viewed as an information system where the actors' main activity relies on accurate negotiation in order to succeed. The methodology is based (1) on the use of a discrete event-based simulator which takes into account features of a cooperation policy as an input, and (2) on a model of the negotiation. This model is built on the semiotic inspired notion of viewpoint which encompasses several essential features of the negotiation, and in particular: the actor, what the actor is interested in, and in what conditions. An algorithm is sketched to design a cooperation policy: It relies upon an iterated process that feeds and manages the simulator along the negotiation taking into account the viewpoint-centred analysis of the results of past simulations.

Keywords: viewpoint, methodology, cooperation, cooperation policy

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 47–62.

3.1 Introduction

When facing a highly competitive market place, companies must focus on and be specialists in their core business. In such a context the number of suppliers and the number of subcontractors tends to increase. Then the realization of some strategic tasks is often ordered to specialized subcontractors [19]. The project managers thus aim at creating reliable partnerships, ideally within a strongly cooperative relationship. Suppliers and subcontractors should be motivated by a mutual interest in such cooperative attitudes. Unfortunately, the project decision-makers often lack tangible arguments. We have explored a pragmatic approach based upon the development of a simulator dedicated to the testing and evaluation of different cooperative decision-making strategies. In this chapter, we study the designing of a cooperation strategy: a semiotic-based notion of viewpoint is used to consider the cooperation strategy as part of the information system made up by the different actors of the cooperation who use the simulator.

After a state of art (section 3.2), we present our own analysis of the cooperative subcontracting relationship within a project supply chain and the simulator that has been built: the idea is to simulate and measure the performances of more or less cooperative policies (section 3.3). Then we sketch a viewpoint-centred methodology to design the cooperation policies (section 3.4).

3.2 State of Art

Project management literature refers to numerous methods presented in the field of project planning and scheduling: project scheduling in make-to-order organizations, deterministic scheduling with resource constraints, resource constrained project scheduling problem (RCPSP), etc. The reader should refer to surveys and reference books, e.g. [7–9, 21]. Nevertheless, project/subcontractor cooperation has not yet been widely studied. Yet, as far as the project is considered as part of a project supply chain collaboration, "… effective collaboration and information sharing are the prerequisites for project supply chain members to succeed" [14]. Due to complexity, most researchers have used qualitative analysis based on quantitative models and concepts drawn from manufacturing management and operation management literature [11–13, 20].

In the field of technological system design and business process improvement (software development process), "the development of complex systems invariably involves many stakeholders who have different perspectives on the problem they are addressing, the system being developed and the process by which it is being developed" [10]. To represent and analyse these different perspectives, viewpoint oriented approaches have been proposed [3–5, 10, 15]. These approaches mostly focus on the software development process modelling and inconsistency management [16, 17]. Viewpoints are also used in enterprise architecture and modelling studies. Frameworks are proposed in order to classify and position the various architectures. Computerized platforms, "which supports the definition, generation, editing and management of architectural views" [18] specified by viewpoints are provided to the architects. Moreover, viewpoint analysis has been applied to the field of concurrent engineering to support team interaction throughout the enterprise [6].

Our study of the state of the art points out on the one hand that few studies exist about cooperation in the project supply chain, and further designing cooperation policies within a project supply chain, though it appears to be one of the most important factors influencing performance of the chain. On the other hand, the "software engineering community appears to have accepted the need to articulate and manage multiple views in the software development process" [10] as well as the enterprise modelling community needs to manage the inherent complexity in enterprise architecture: viewpoint approaches and frameworks have then been proposed. Viewpoint-centred approaches have been mostly investigated in systems designing and software development or enterprise modelling when many stakeholders are involved within a single enterprise. In this chapter we aim at evaluating the interest of viewpoints to design cooperation policies between actors belonging to different enterprises of a subcontracting relationship within a project supply chain.

3.3 Cooperative Subcontracting Relationship: A Simulation Approach

3.3.1 Process analysis

In order to improve the intelligibility of the decision-making process, we have studied industrial cases involving a resource centre specialized in aerodynamics: ONERA-Fauga is a specialized centre in aerodynamic testing facilities. The central resource is a compressed air system and an array of industrial wind tunnels. Each year different projects used to schedule their aerodynamic tests in this centre. One of the case studies concerns

Airbus and ONERA-Toulouse. Two projects use this strategic resource in their activities in order to test and validate aerodynamic models.

The resource reservation process

The resource belonging to the subcontractor is shared between two projects. So each project manager has to schedule the tasks of his project according to the time-slots available for this external resource: these time-slots are called the *time windows*.

When scheduling his testing tasks, each project manager – i.e. from Airbus and from ONERA-Fauga – mainly relies on the current state of his activities, and especially on his current schedule. He specifies his requirement on this basis before contacting the subcontractor (ONERA-Fauga). A more or less formalized dialogue is then initiated between the two project managers in the framework of the project/subcontractor relationship. Each project manager has working knowledge of the needs, capabilities, and communication strategy of the partner. The dialogue ends up with the reservation of a time window (see Fig. 1). Then each project manager integrates the time window in his own schedule which is modified accordingly. The quality of the cooperation in the project/subcontractor relationship is directly related to the quality of the information transmitted by the project managers: it obiviously depends on how information is used by both of the project managers.

The updating process

In this kind of medium time term project, the reservation is only provisional and set up far before the realization of the task, i.e. several months before.

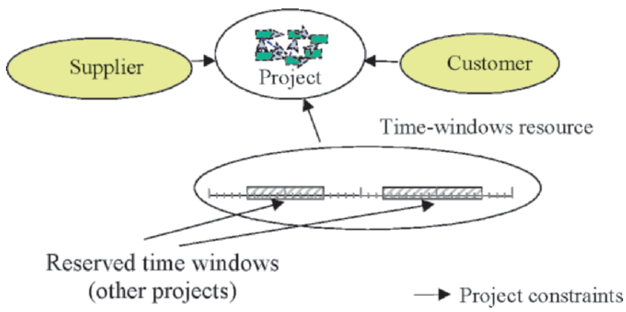

Fig. 1. “Time windows” resource

During this period, perturbations occur whose impact on the activities of each project manager can be severe: these perturbations can invalidate the time window originally reserved. These time windows must be brought up through a process of updating that runs until the final execution of the task. Both project managers interact during this period in order to maintain valid time windows. If needed, the reservation is updated and the schedule is modified accordingly.

The updating process can be viewed as a series of reservation processes. The maintaining of the time window takes place periodically, and it depends on the rescheduling period of the project managers, or on demand, when significant factors impact on the resource or on the project.

3.3.2 A simulation tool

We are interested in determining the "best" policies of cooperation in the context of a project/subcontractor relationship. Obviously such policies cannot be automatically generated from a model of the relationship – this would presuppose that it is possible to define explicitly the optimization criteria which govern the relationship. But such criteria are unknown. So we propose an evaluative approach that measures the performance of the cooperation policies chosen by the two actors – the project managers – of the project/subcontractor relationship. The measurement includes the development of a temporal indicator that evaluates the risk factor of a cooperation policy. The respect of delivery dates is indeed one of the most impacting factors both in project and resource management. The performance is not only measured at the end of the initial reservation process but also after each update of the schedules. In order to perform this evaluation, we have already specified and implemented a prototype tool which simulates the dynamics of the relationship between a project entity and a subcontractor entity. The prototype is based on a discrete event simulation. The idea is to track the evolution of the project/subcontractor relationship over a given horizon. The targeted users of the simulator are a project manager and a decision-maker in charge of the subcontractor resource. It relies on a triple model: a model of the project, a model of the subcontractor – the resource – and a model of the relationship that organizes their interaction – namely, the reservation and the update of a time window on the resource maintained by the subcontractor. The users will define the parameters for each of the components of the model (cf. Fig. 2).

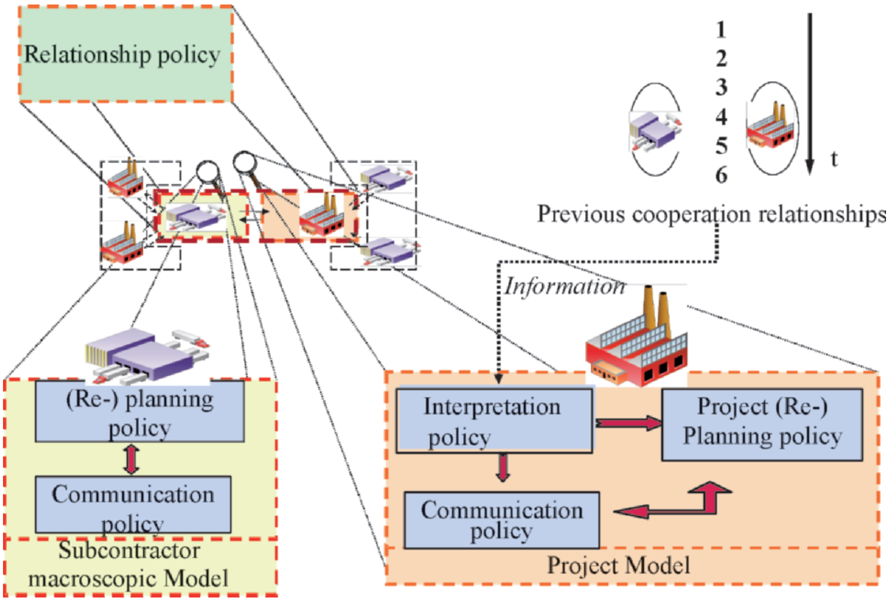

Fig. 2. The main organization scheme of the simulation tool

The simulation tool provides both actors with a means of testing their policies of cooperation in the context of the time window reservation process. In our approach, the notion of "cooperation policy" includes the information exchanged by the actors and the way information is processed on both sides, and more particularly the way the information is integrated into the schedule.

3.4 A Viewpoint-Centred Methodology to Design a Cooperation Policy

3.4.1 Viewpoint-centred designing

Two key points give rise to the importance of the concept of viewpoint in cooperative designing: the viewpoint concept is central to two processes, the process whereby actors who cooperate in the process aimed at designing an object communicate amongst each other and the process whereby this object achieves sense [1].

The act of designing a new object (an industrial system, a software component, etc.) brings into play a great many technical, organizational, and financial skills to find solutions to problems such as architecture, technical constraints, signal transmission, controlling cost prices, managing and coordinating teams … over a period that may last for a very long time. It can be said that all actors contributing one of these skills to the project has his or her "own" object that must be integrated with that of his or her partners.

The quality of communications between actors is therefore a first key point to the project's success. Indeed, in this cooperation activity, the actors are exchanging partial, incomplete, and even contradictory information. The basic idea is to no longer only take into account the representations of the future "object" but also the object itself, its design process, and the transitory conceptions by each actor in all their manifold complexity.

The second key point is to take into consideration the sense of the object and the actors that give this object sense. In this way, the object to be designed and an actor participating in the design project are not isolated entities: the object gets sense when it is connected to how it is interpreted by an actor. Any representation of an object is thus subjective and contextual.

This position is conductive to a systemic view of objects, actors, representations, and the design process: the object only exists when it has acquired sense for all the concerned actors. The object's sense is then also the result of the designing process of the object.

Let us take the following toy example [2]: designing a new car (cf. Fig. 3). Two actors a and a' decide to design a new car. The designing process creates an information system the kernel of which are the actors a and a'. Let us sketch the designing process. At first designer a communicates his conception – idea – to the other designer a' by means of an expression e, which is made of words. Now a' receives message e from actor a as she conceives it as an expression e' which denotes a conception c' of her own. Then a' sends back this expression e' to a in order to verify his understanding of a's conception. e' denotes another conception c'' for a. Now, for a, if $c = c''$, a mutual understanding is reached and the design is successfully achieved. If not, the communication goes on between designers a and a' until agreement is reached on the identity or no mutual understanding. When an agreement is reached, it relates to the sole expressions, the concepts remain individual.

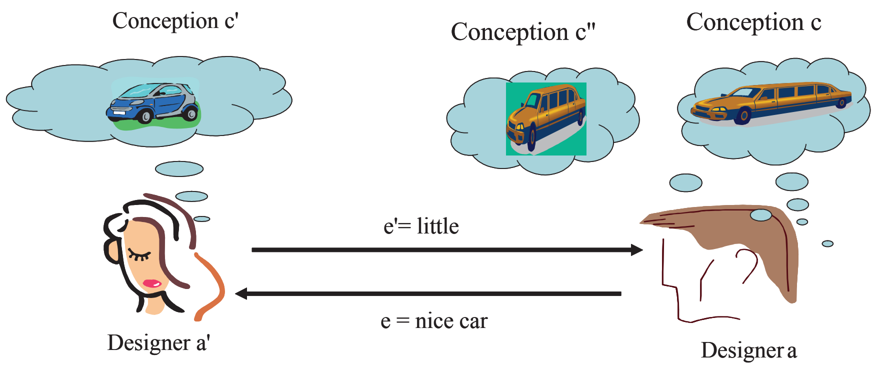

Fig. 3. Two designers, *a* and *a*′, with their own conception (idea) *c* and *c*′ of a car

3.4.2 Viewpoint definition

Let V = <*A*,*O*,*S*,*E*,*C*,*V*> be an universe of viewpoints such that

- A = a set of actors $a_1, a_2, \ldots, a_i$;
- O = a set of objects $o_1, o_2, \ldots, o_m$;
- S = a set of contexts $s_1, s_2, \ldots, s_j$;
- E = a set of expressions $e_1, e_2, \ldots, e_l$;
- C = a set of concepts $c_1, c_2, \ldots, c_k$;
- V = a relation between A, O, S, E, and C;

where i, j, k, l, and $m \in \mathbf{N}$.

Accordingly, the universe of viewpoints is a Cartesian product $V = A \times O \times S \times E \times C$.

A particular viewpoint is then denoted as $V(a,o,s,e,c)$. Geometrically a particular viewpoint can be described as a hexahedra having five vertices and nine edges between them [2, 3] (cf. Fig. 4).

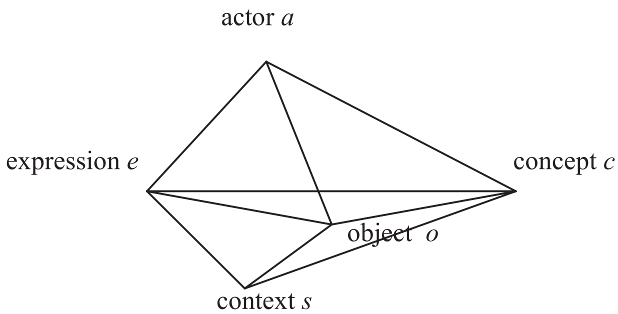

Fig. 4. A viewpoint as a geometrical hexahedra

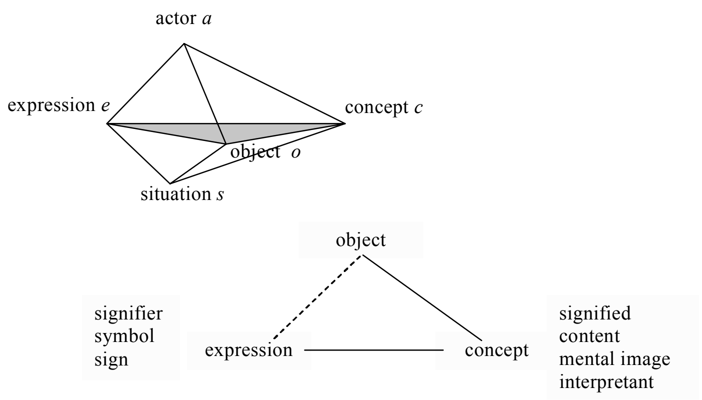

Fig. 5. Viewpoint with semiotic triangle

3.4.3 Semiotic interpretation

Viewpoint

A viewpoint comprises Peirce's sign triad [3], the actor, and the situation: who produces the sign and in what conditions the sign is produced (cf. Fig. 5).

Subrelations inside a viewpoint

Among the subrelations, we distinguish those which imply concepts and thus carry on sense, the meaningful subrelations, and those which do not implicate concepts, the meaningless subrelations. Here are interpretations of some of the subrelations involved in a viewpoint.

Meaningful subrelations

I(a,s,c): "idea" relationship, for instance the starting point of designing where *a* has a concept *c* in a context *s*

C(*a*,*s*,*c*,*e*): intentional expression of the *idea* by *a*

S(*c*,*e*,*o*): semiotic *sign* triangle (Peirce)

V(*a*,*o*,*s*,*e*,*c*): extensional expression, a viewpoint here

Meaningless subrelations

E(*a*,*s*,*e*): communication of expression *e* by designer *a* in situation *s*

Design process

The design process itself can be observed as a series of transitions and transformations between meaningful and meaningless subrelations. Two levels of observations can be put forward: the individual level of each actor, and the general level of the collective designing process.

Individual designing process schemata
I(*a*,*s*,*c*) → *conceptualization* → *C*(*a*,*s*,*c*,*e*) → *objectivation* → *V*(*a*,*o*,*s*,*e*,*c*)

Collective designing process
V(*a*,*o*,*s*,*e*,*c*) → *conceptualization* → *C*(*a*,*s*,*c*,*e*) → *communication* → *E*(*a*,*s*,*e*)

When an agreement is reached, it relates only the expressions, the concepts remain individual.

3.4.4 Viewpoint-centred cooperative designing process principle

Let us use the previous subrelations to describe the designing process sketched in section 4.1 above between the two actors *a* and *a´*.

```
1- a in a situation s has an idea c, i.e.I1(a,s,c)

2- a gives it an expression e, i.e. C1(a,s,c,e)

REPEAT

  3- a´ in situation s receives a's expression e,
  a´ receives it as e´, i.e. E1(a´,s,e´)

  4- a´ conceptualizes it as a c´,
      i.e. I2(a,s,c´) ⊆ C2(a´,s,c´,e´)

  5- a´ communicates, in turn, her expression e´
  to a.

  6- a receives a´'s expression e´ as an expres-
  sion e", i.e. E2(a,s,e") which designer a has to
  conceptualize as I3(a,s,c") ⊆ C3(a,s,c",e")

  7- a compares conception c" with c, whether
  I1(a,s,c) is similar to I3(a,s,c") or not.

  8- IF c is similar to c"
        THEN an agreement is reached (c is similar
  to c')
        ELSE
```

```
      a has to adjust and communicate his con-
  ception to a´ by means of expression e´´´,
      a' conceptualizes the received expression
  e´´´´, I4(a,s,c'''),and compares it to her own
  earlier conception I2(a,s,c´).
UNTIL the similarity between the conceptions is
reached, or a and a' decide to stop designing
```

3.4.5 Cooperation policy

We define a cooperation policy as the instantiation of actors' policies and behaviours among a set of potential policies and behaviours. In a first step, we focus on a set of potential policies and behaviours of the actors and some properties of the relationship between the two actors which are implemented in the simulator.

The relationship is characterized by the strength position between the two actors.

Project behaviour
It is characterized by:

- The information reception behaviour, i.e. the interpretation of the information provided by the subcontractor;
- The project planning/replanning policies;
- The communication behaviour, i.e. transmission of data to the subcontractor;
- The degree of uncertainty pertaining to the project and likely to generate hazardous events.

Subcontractor behaviour
It is characterized by:

- The subcontractor planning/replanning policies;
- The communication behaviour, i.e. transmission of data to the project manager;
- The degree of uncertainty pertaining to the resource and likely to generate hazardous events.

3.4.6 Cooperation policies and viewpoint definition

As far as a cooperation policy is concerned the universe of viewpoints can rely upon the following instantiation:

- A = {subcontractor decision-maker, project decision-maker, tool designer}
- O = {actors policies and behaviours, relationship characteristics}
- S = {moment-place}
- E = {speaking language, parameter choice, algorithm, mathematical model}
- C = {confidence, temerity, taking the other actors constraints into account, relaxation of one's own constraints}
- V = a relation between A, O, S, E, and C

Examples

Here are three examples of viewpoints met in a cooperation policy:

1. <
Actor: subcontractor decision-maker
Object: relationship characteristics
Context: {24/06/05, ONERA}
Expression: {tool parameter concerning the position of strength in the relationship: *ST>PJT*)}
Concept: {the subcontractor is in a position of strength}
>

2. <
Actor: project decision-maker
Object: communication behaviour
Context: {24/06/05, ONERA}
Expression: {tool parameter concerning the communication behaviour: 0%}
Concept: {very confident in the project planning policy}
>

3. <
Actor: simulation tool designer
Object: project behaviour
Context: {4/06/05, ONERA}
Expression: {Mathematical model: the "possible time windows" Cp_i which are considered for the planning process are computed according to the transmitted time windows by the subcontractor}

$$Cp_i = \bigcup_{y \in [1,m]} \left[Ep_y^i , Sp_y^i \right] \qquad (1)$$

with

$$Ep_y = f(A_1,..,A_n,B_1,...,B_n,50\%)$$
$$Sp_y = f(A_1,..,A_n,B_1,...,B_n,50\%) \qquad (2)$$

Concept: {the project manager is very suspicious: overestimation (50%) of the free time windows is suspected}
>

3.4.7 A viewpoint-centred method for cooperation policies design

A viewpoint-centred method can be defined in order to design a “good” cooperation policy and to update the parameters of the simulation tool. The following algorithm summarizes this method:

REPEAT

```
  The subcontractor expresses a set of viewpoints
  The project expresses a set of viewpoints
  Construction of an experimental design
  (N cooperation policies i.e. N combinations of
  viewpoint expressions)
  IF the cooperation policy can be tested with the
       simulation tool
   THEN Evaluation with the simulation tool
   ELSE
      The tool designer expresses his viewpoint on
  the concerned object (which expression cannot be
  tested)
      Confrontation of the different viewpoints in
  order to express a new behaviour or policy
      Re-design of the simulation tool with the
  integration of the new expression
UNTIL both actors are convinced of the cooperation
policy
```

Figure 6 illustrates this algorithm.

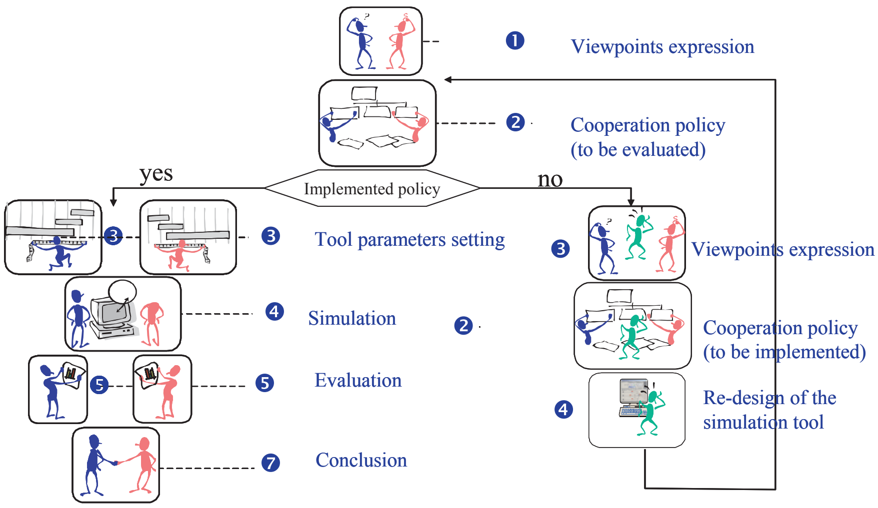

Fig. 6. The viewpoint-centred design of the cooperation policy

3.5 Conclusions and Future Research

We have presented in this chapter a methodology that aims at improving management projects based on a relationship project/subcontractor and a common resource. In our approach such a project is viewed as an information system where the main activity of the actors relies on accurate negotiation in order to succeed. The methodology is based upon the use of a first stage discrete event-based simulator which takes into account features of a cooperation policy as an input, and on a second stage model of the negotiation that relies upon the semiotic inspired viewpoint notion. The main interest of the viewpoint notion is that it takes into account several essential features of the negotiation: the actors, what they are interested in, and in what conditions. An algorithm is sketched to design a cooperation policy: it is based upon an iterated process that feeds and manages the simulator along the negotiation by means of a viewpoint-centred analysis of the results of past simulations.

The method will be applied in the context of the cases already used to evaluate the simulator.

References

1. Charrel PJ (2002) The viewpoint paradigm: a semiotic-based approach for the intelligibility of a cooperative designing process. Australian Journal of Information Systems, 10(1), pp. 3–19
2. Charrel PJ, Palomäki J (2005) Viewpoint Centric Design, Workshop Philosophical Foundations of Information System (Marcos E and Wierringa R (eds.)). In: Proceedings of the CAISE Workshops, Vol. 2 (Castro J and Teniente E (eds.)), FEUP Edicoes, pp. 475–481
3. Charrel PJ (2004) An information system designing as a persistent multi viewpoints requirements elicitation process. In: Liu K (ed.), Virtual, Distributed and Flexible Organizations. Studies in Organizational Semiotics, Kluwer Academic, Dordrecht, The Netherlands, pp. 103–118
4. Finkelstein A, Kramer J, Nuseibeh B, Finkelstein A, Goedicke M (1992) Viewpoints: a framework for integrating multiple perspectives in system development. International Journal of Software Engineering and Knowledge Engineering, 2(1), pp. 31–58
5. Finkelstein A, Sommerville I (1996) The viewpoints FAQ. Software Engineering Journal, 11(1), pp. 2–4
6. Hanneghan M, Merabti M, Colquhoun G (2000) A viewpoint analysis reference model for concurrent engineering. Computers in Industry, 41(1), pp. 35–49
7. Kolisch R (2001) Make-to-Order Assembly Management; Springer, Berlin
8. Kolisch R, Padman R (2001) An integrated survey of deterministic project scheduling. Omega, 29, pp. 249–272.
9. Neumann K, Schwindt C, Zimmermann J (2003) Project Scheduling with Time Windows and Scarce Resources: Temporal and Resource-Constrained Project Scheduling with Regular and Nonregular Objective Function, Springer, 1 August
10. Nuseibeh B, Kramer J, Finkelstein A (2003) A Viewpoints: Meaningful Relationships are Difficults, Proceedings of the 25th International Conference on Software Engineering, Portland, Oregon
11. O'Brien WJ, London K, Vrijhoef R (2002) Construction Supply Chain Modeling: A Research Review and Interdisciplinary Research Agenda, IGLC-10, August 2002, Gramado, Brazil
12. O'Brien WJ, Issa R, Hammer J, Schmalz MS, Geunes J, Bai SX (2002) Seek: accomplishing enterprise information integration across heterogeneous sources, http://www.itcon.org/2002/7, Electronic Journal of Information Technology in Construction, 7, pp. 101–123
13. Palaneeswaran E, Kumaraswamy M, Thomas S (2003) Formulating a framework for relationally integrated construction supply chains. Journal of Construction Research, 4(2), pp. 189–205
14. Siniharju M (2000) Collaboration in Project Supply Chain, TU-22.451 Seminar in International Project-Oriented Business, Helsinki

15. Sommerville I, Sawyer P (1997) Viewpoints: principles, problems and a practical approach to requirements engineering. Annals of Software Engineering, 3, pp. 101–130
16. Sommerville I, Sawyer P, Viller S (1999) Managing process inconsistency using viewpoints. IEEE Transactions on Software Engineering, 25, pp. 784–799
17. Spanoudakis G, Zisman A (2001) A inconsistency management in software engineering: survey and open research issues. Handbook of Software Engineering and Knowledge Engineering, 1, pp. 24–29
18. Steen MWA, Akehurst DH, ter Doest HWL, Lankhorst MM (2004) Supporting Viewpoint-Oriented Enterprise Architecture. Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference, Monterey, CA
19. Thierry C, Lamothe J, Galvagnon V (2004) A re-planning support system to small batch production with booked resources. International Journal of Production Research, 42(23), pp. 4993–5008
20. Vrijhoef R, Koskela L (2000) The four roles of supply chain management in construction. European Journal of Purchasing and Supply management, 6, pp. 169–178
21. Weglaarz J (1998) Project Scheduling – Recent Models, Algorithms and Applications, Kluwer's International Series

Part 2
Risk Management

Chapter 4
A Contribution to a Semiotic Approach of Risk Management

D. Galarreta

Centre National d'Etudes Spatiales,
18, av. Edouard Belin, 31401, Toulouse cedex 9. France

Abstract

In this chapter we examine how a semiotic approach of risks can be proposed and how the concept of affordance can be adapted to such a goal. We are reminded of the primitive notion of action and its close relation with risks. Perception issues are examined in order to make clear the relation between the concepts of action and of affordance. It turns out that the affordance concept does not belong to the primitive action paradigm and a risk cannot be entirely described as an affordance. A multi-viewpoints semiotics offers a convenient framework for defining a risk as a semiotic concept. We examine the question of managing risks in the special case of the Rosetta long-duration mission to prevent uncontrolled knowledge evolution. It appears that managing risks of knowledge evolution should be based in this case on the combination of text-mining techniques and organisational arrangements.

Keywords: multi-viewpoints semiotics, affordances, risk management

4.1 Introduction

The complex character of the organisation of large projects gives a new vision about the risk notion. If we consider for instance the risks of material or corporeal damages in large projects, two different approaches exist. The

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 65–88.

first one is called the High Reliability Approach. It considers that accidents can be avoided provided good organisation and good management are implemented. The other approach is called the Normal Accident Theory. It considers that accidents are inevitable in complex systems. Here complex system means a system of behaviour which cannot be explained by only one point of view or theory but requires several viewpoints in order to understand it.

In this chapter we will try accordingly to argue that risks are not only inevitable but also necessary to better apprehend the technical objects which are designed or used in such contexts. Instead of being a sign of ignorance, risks correspond on the contrary of what is usually defined as *a piece of knowledge*. The impression of being in front of a paradox stems from the fact that knowledge is usually associated to a positive element although a risk often appears as a limit beyond which it is dangerous to venture.

However one motivation in this chapter is to examine the concept of affordance and how it can deal with the question of risks. This examination leads us to study the primitive notion of action and its close relation with the risk notion. In order to support this programme we refer to philosophers of action and of perception (B. Saint-Sernin, M. Dufrenne, and M. Pradines) and of course to J.J. Gibson. Ideas have a history. Their novelty does not change that fact. Referring these ideas to older ones that we ignored or forgot could be a way to perceive the true originality of the new ones. We hope that the reader will forgive us for long quotations of authors who are usually ignored by the usual audience of organisational semiotics, but whose works belong to that history of ideas. This chapter's plan is as follows:

- We first consider the genesis and the unity of the primitive concept of action.
- Then we evoke M. Pradines's conception of action. M. Pradines proposed a view of the sensation concept which both agrees with the notion of action we present and is close to the approach of J.J. Gibson for perception.
- We then analyse the perception theory of J.J. Gibson and its relation with action.
- We show how J.J. Gibson and the organisational semiotics derived the concept of affordance from perception ideas; and how they deviate from the action primitive paradigm from where they stemmed.
- We analyse to what extent a risk can be considered as an affordance.
- We examine the concept of risk within a multi-viewpoint semiotics.
- We end our chapter by examining the special case of risks of knowledge evolution in a long-duration space mission such as the ESA's Rosetta Mission.

4.2 Action and perception

4.2.1 Genesis and unity of action

According to Bertrand Saint-Sernin “there is an action if and only if one or several persons by their behaviour bring about a modification of the outside world. A project or an intention which remains in the form of an idea does not constitute an action. It only constitutes a thought. However there is not an action when a change of the outside world is not the result of a project. If by chance, without any intention I cause a spate of events, it does not imply that I have performed an action: this is precisely what we call an *accident*” (Saint-Sernin 1989, p. 15).

This conception stemmed from the Greeks, first through Homer, then through Plato and Aristotle. “The inventors of the Greek archetype of action understood first that God should not be considered as responsible for our business and our misfortune. They saw that his innocence coincides with the freedom of man. They perceived that in order to be applied to great tasks, this freedom had to be demonic, that is singular, inspired and, but impersonal and rational. They understood that action had a goal if not unique at least a privileged one, the safety and the preservation of the cities, that is politics” (id., p. 10). This conception was then adopted and adapted by the Judeo-Christian tradition (ibid., pp. 10–14).

This conception of action should be distinguished then from the notions of gesture or of act. “An action should not be the execution or the reproduction of a gesture. Does the pole vaulter who takes a run up for the thousandth time, or the parachutist who throws himself out of the plane for the hundredth time, perform an action? … The gesture as such is not sufficient to constitute an action as far as it illustrates by repeating it a model which belongs to an established institution. … What distinguishes an *act* from a gesture is that it does not rest on an automatic execution, but implies the intervention of mind, the application of the whole being to its realisation” (ibid., p. 20). “One cannot say either that a singular act constitutes in itself an action: an act has a beginning and a completion. It is datable and delimited. …An action has often a beginning which is difficult to ascribe and the incompleteness of it is almost intrinsic” (ibid., p. 21).

“By convention, we will reserve the name of *action* to enterprises which extend over time, which involves risks and which produces effects which are both wanted and unpredicted upon the world. An action necessarily has initiators, but it has not always ascribable authors, because through the passage of time individuals hand over to one another.” (ibid., p. 22)

The question of the agent’s freedom that B. Saint-Sernin mentions, in the classical acceptation of action, needs to be specified. It can appear difficult to evaluate this freedom because of the constraints borne by the agents existing outside as well as inside themselves. Actions appear "only if motives exist in those who start them and in those that these actions make them move. These motives are of two kinds … vanity, ambition, self-esteem or altruism, quest of good, sense of duty, etc." (p. 18) "voluntary adherence, agreed obedience, forced submission are mixed in the performance of action" (ibid., p. 18). How then can we evaluate the true freedom of an agent?

In order to escape this difficulty, we can alternatively suggest to reduce this freedom to the quality of an agent who would fully be a man of action, that is who would fully realise his nature as man of action.

Among the features which characterise the latter, there is an attention paid to the right moment, the *kairos*; this practical intelligence in opposition to the theoretical intelligence, i.e. of the pure forms (ibid., p. 23). A situation can be seen as either an object of vision or an object of action. Confusion leads to errors and failures. This practical intelligence is directly related to the capacity of the man of action to give significance to a situation while being confronted with things or people.

"The man of action is often the one who reacts towards the known as if it was unknown and toward the unknown behaves as if it was familiar to him. In the first case he sees with a fresh eye what others would treat as too well known; in the second case, he acts with ease towards the unknown. It is this unity of intention, of inspiration, more than the individuality of the agents which contributes to the consistency of an action." (ibid., p. 24)

"The essence of action is characterised by an element which is irreducible to the well known, the sound and the experienced. Its specificity is precisely the assumed risk, the desired invention, the accepted unexpected." (ibid., p. 24)

"The essence of action is characterised by an element which is irreducible to the well known, the sound and the experienced. Its specificity is precisely the assumed risk, the desired invention, the accepted unexpected." (ibid., p. 24)

"Rightly or wrongly, the author of a work thinks that he controls the development of it whereas the performer of an action knows that he must, permanently, take support, to go further, in the circumstances. He needs the bearing forces of the things." (ibid., p. 23)

While doing so, he solves a problem which is not general and abstract but specific and concrete and the situation characterised in this way gains a supplement of identity. The value system from which he gets moral, symbolic and material resources for his action is reinforced and sometime updated in this occasion. The corresponding competence that he needs to build the identity of that situation and to use this value system at his disposal is a semiotic one because that building is a semiotic operation just as using a value system implies a semiotic competence (see Galarreta 2004).

Moreover, "allowing for an exception, action is always interaction. The stage on which it is performed is populated by agents whose behaviours interfere" (Saint-Sernin 1989, p. 25).

Saying that the action consists of semiotic doing of a free agent in situation of confrontation with other agents, both results from its traditional conception (as we tried to show it) and also precise its nature.

4.2.2 Pradines's conception of action

This conception leads us to evoke a philosopher who is today somewhat forgotten. He developed a theory of perception and a theory of knowledge in which the concept of action follows the conception we have just introduced. It is Maurice Pradines.[1]

The definition that Pradines gave of an action in his thesis dissertation in 1909, brings to the fore that what he defined as an instinct to reduce the multiple to the unity (the purest manifestation of which is thought), is a semiosis operation.

"Any action is a relationship between two things, that which acts and that which undergoes the action. If there is, in this relation a mystery, it is the universal mystery, given in any phenomenon; we do not acknowledge another in the phenomenon of thought. To live, in particular, is to undergo actions in the form of multiple impressions, and to re-act, to preserve ourselves against them or by their means. Therefore the simplest life already presents to us both the opposition and the interdependence of a sort of living unit and a living multiplicity, the former trying to reduce the latter: this effort is only the very instinct of living. We believe the act of thinking is simply the highest form of this instinct; the effort of the thought to reduce the multiple to the unit is the most perfect demonstration of the need which animates one to preserve oneself against others. We believe that this effort constitutes any thought, which leads to the following: there is only one category of mind, the unity, and any thought is, basically, mathematics or rather arithmetic; consequently there exists, no difference as for the operation of the mind, between the laws of the physics and the principles of mathematics; any knowledge is synthesis, any synthesis is empirical, one can find the perfect necessity in syntheses of experiments, and one cannot

[1] Maurice Pradines (1874-1958). He was thesis director of Emmanuel Levinas in 1930. Further information on his philosophy of sensation can be found in (Guendouz, 2003)

find it out of these syntheses, since one can think only by their means." (Pradines 1909, p. 26)

Let us remember that there too the conception of the action which is presented conflicts with another conception of the action as one distinguishes it from intelligence or thought: either conceived like an order of faculty differing radically from the representation and opposed to this, or conceived as being what surrounds intelligence, preceding and preparing it, following and transcending it (see article *action* p. 20 in Lalande 1985). In this conception the thought operates on "mental" representations whereas in the other conception (that of Pradines) the mind tries to reduce a gap between inside and outside, "a kind of living unit" trying to reduce "a living multiplicity". In this case the operation of the thought does not require any more a priori mental representations since it "externalises" partly the resources which enable it to be exercised. It is important at this point to remember that in its recent developments, semiotics of the discourse insisted on the fact that "the signification supposes ... a world of perceptions, where the proper body of the operator (sensitive envelope) by taking a position in the world, installs two macro-semiotics (the natural language and the natural world) and whose border can always move, but which have each one a specific form ... the signification is thus the act which joins together these two macro-semiotics, and this, thanks to the proper body which has the property to belong simultaneously to two macro-semiotics between which it takes a position" (Fontanille 1998, p. 35). "Semiosis is proprioceptive" (id., p. 41).

4.2.3 A semiotic definition of action and how it relates to the risk notion

Let us sum up what we have set out. We have explained the concept of action that stemmed from Antiquity which is characterised by the freedom of its agents. This condition implies that action is a semiotic doing or more precisely *that the action consists of a semiotic doing of a free agent in a situation of confrontation with other agents*.

Pradines's conception of perception rests also on a semiotic concept of action (even if it addresses the elementary level of sensation).

We pointed out that Pradines's analysis and recent developments in semiotics lead us to believe that perception can exist without pre-existing mental representations since it rests partly on an externalisation of the resources it needs. We can maintain the notion of *images* in the semiotic processes we consider, provided we mean *sensitive images*, i.e. involving the proper body of an agent, and whose meaning does not exist prior to semiosis.

Therefore following Pradines we can complement the above *definition of action* in the following way as:

Thinking without mental representation but with sensitive images which consist in a semiotic doing of a free agent in situation of confrontation with other agents and the outer world.

Conversely, does any semiotic doing performed agent in situation of confrontation with other agents and the outer world correspond to an action?

To be more specific let us take a simple example. Can we consider that an individual reading a text is performing an action in the usual acceptation of it? Do I perform an action by reading an Agatha Christie's novel? I certainly perform a semiotic undertaking but not an action in the usual acceptance of the term. Therefore any semiotic task is not an action. One can point out that this reading situation is missing confrontation with other agents. But this in turn leads us to specify the *confrontation* notion we implicitly use: indeed reading a book implies being connected to a socio-cultural group. Therefore *confrontation with other agents* is not necessarily a *connection with other agents*. What does connection miss? In order to give a hint let us imagine our reader reading a handout forbidden by the police (in a totalitarian society). Reading in this case implies *risk taking*. Let us remark that: the intensity of the risk of reading a forbidden message is not so much related to the handling of a forbidden handout as to the adherence of the reader to the criticisms of the text. If I am unable to read it or if I totally disagree with its content, the risk of my reading would appear to me to be less serious.

A man of action is someone who takes risks. He can take risks either by reacting to outside actions or by refusing to react to these actions. In both cases the significance of the adopted attitude by the agent includes its risk component. The presence of a risk will be experienced all the more intensively since the agent is more decided in performing the corresponding action.

Similarly we can postulate that the expanse of the risk component will be experienced all the more clearly since the agent is more decided to perform the corresponding action. The risk component is the collection of individual risks that are experienced when an action is undertaken.

In fact the characterisation of a semiotic doing as an *action* is all the more asserted since the risk is more intensively and extensively experienced by the agent. By adopting this position we make risk a semiotic concept.

4.2.4 We now show the consequence of this type of thought and how it anticipates the propositions of Gibson

Pradines is a philosopher of sensation. When he considers sight for example, it appears to him that "the eye explores the world by the foveal activity which is the copy of the activity deployed by the fingers. Sight thus copies tact" (Dufrenne 1987, p. 22). When he compares feeling to perceiving he also notes "that instead of causing an immediate and blind reaction, the impression exerted on an organ expresses a quality. However this impression 'matters only by its expressive' quality; it alerts the living only if the livings it, only if the living understands the imminence of a defined vital action, conditioned by the movement of an agent'. In other words even if this quality correspond to an inner state caused by an organ and a nerve (cf. Müller's law), it cannot be reduced to it: this quality also describes an object and keeps it at distance. Subjectively, the impression is objectified, the internal, reaction becomes externalized ... the sensation does not lead the living to withdraw into itself, it brings the living into the world, and it informs of the businesses of the world. The sensation means at least that the living become a subject; it becomes aware of an object apart from him. Apart from him, should be taken literally: the contact is broken or rather anticipated, the object is parted from the representation. This representation is not 'intellectual', it informs the subject on what can touch it: 'what the sensation represents, is always a possible affection and the object which causes it'; but it allows the subject to be informed without being touched. In the word representation, the *re* means the carving of space, but it signifies by no means that space is only representation, that conscience only knows its states. The representation does not suppress presence; it does not conjure away the world" (id., pp. 23–24).

"The movement cannot be perceived just by itself. Remove the conscience of the difference, as it results from the contact: you will be able to preserve the movement but you will suppress its perception.

It is the reason why a homogeneous space is unknowable. … The concept of place results from the composition of the movement and of the contact, and one can see that its origin is very practical" (Pradines 1909, p. 54). "It is thus the stopping of the movement which creates, for us the sensation of what is not us. This object (objectum) is situated in a place by definition of word, but by no means by its nature. This opposition indicates exactly the place. But the opposition is born from the collision of two actions, and, consequently, the place is a product of the action and by no means an intuition, neither a priori, nor primitive" (id., p. 55).

We analyse now the theory of Gibson to show how it rests on a conception of action which is close from Pradines's and consequently on the conception initially introduced.

"When light is many-times reflected from an array of surfaces – when it 'fills' the environment as we say – it has the unique property that reflected rays converge to any point to the medium. The objective environment is projected to this point. If an eye is placed at that point it can register a sector by the familiar process of the formation of an image" (Gibson 1958, p. 183). The rays converging at this point will have different intensities (and frequency compositions) in different directions, and they constitute what is termed by Gibson, an optic array.

Gibson made several points that play an important role in our argument.

He remarked that: "we have generally believed that only the focused light constituting the retinal image excite the receptors. But physiological conception of the stimulus has been a source of paradox and confusion in psychology. In fact it does not apply. The image is a stimulus for an eye, which responds first by focusing it. The image is no more than a response-produced stimulus. A retinal image is not a thing with definite boundaries in any case. The retina continually moves behind it, with both large and small incursion, so as to bring the fovea to different bits of details" (id., p. 184).

Let us call this remark "a solution to the elementary sensation/perception dichotomy issue".

Then he argued that an eye of an animal is not only sensitive to static patterns but also to the flow patterns when this animal is in movement relative to its environment. This ability makes a new sort of kinaesthesia (i.e. sensitivity to different kinds of motion) possible, which he defined as a visual kinaesthesia:

"An eye is a device which registers the flow pattern of an optic array as well as the static pattern of an array. Conversely, such a family of continuous transformations is a stimulus for an eye. There are quite specific forms of continuous transformation, and the visual system can probably discriminate among them … This mode of optical stimulation is an invariable accompaniment of locomotive behaviour and it therefore provides 'feedback' stimulation for the control and the guidance of locomotive behaviour. It might be called visual kinaesthesia.

The last assumption asserts something like an unrecognised sense of modality. Visual kinaesthesia is, of course, supplementary to the recognised mode of proprioceptive kinaesthesia. It differs, however, in several ways. Firstly, it seems to provide information about movements of the animal relative to the environment, not about movements of parts of the body relative to other parts, as the muscle-sense does. Secondly, it seems to provide information about displacements rather than information about acceleration and gravitation forces, as the vestibular sense does. Thirdly, the displacements registered have reference to the stable solid surfaces of the environment; displacements with reference to the medium of air or water, in the case of flying or swimming animals, are given only by proprioceptive kinaesthesia. Kinaesthesia has long been defined as the sense of bodily motion … It depends on the sensitivity of the receptors in the muscles and joints to compression, on the sensitivity of statocyst to force, and also to the sensitivity of the skin to deformation. Visual kinaesthesia depends on the sensitivity of a retinal mosaic to an overall change of pattern." (ibid., p. 185)

At this point we will emphasis the fact that *visual kinaesthesia* combines at least a visual stimulation with a locomotive expense and effort.

"Animals make different kinds of locomotive reactions to different objects. They approach food or shelter, they avoid obstacles, they pursue prey and they flee the predator. These are discriminative reactions and they require a different kind of stimulus-response theory than do the control reactions heretofore considered. We must now consider actions which are specific to those features of the optic array which do *not* change during locomotion rather than those which do. Such features of stimulations are not response produced and the responses are not circular. In such behaviour the S-R linkage is between permanent entities of the environment and acts which are appropriate to them. The distinction between an S-R theory of control *reactions* and an S-R theory of *identifying* reactions is important for behavioural theory. It is true that an automaton can be designed which will aim at, approach, and pursue a pre-set target (as witness military missiles) and that no automaton has yet been designed that will recognize targets appropriate to its own needs (apart from its designer's) and act accordingly. But it would be wrong to categorize the first kind of reaction as *automatic* and the second kind as *voluntary*. This dichotomy is as pernicious as the one between sensory and perceptual processes. The true distinction is probably between the properties of stimulation which vary over time and those which do not." (ibid., p. 190)

Let us here underline the "S-R linkage is between permanent entities of the environment and acts". As in the solution to the elementary sensation/perception dichotomy issue, where the retina movements permit to avoid the dichotomy between sensory and perceptual processes, it is internal transformations of the animal (namely muscular movements to adapt itself to the "object"), which permit us to escape the "pernicious dichotomy".

4.3 Affordances

“At the origin, the concept of affordance emerged from the works of Gibson in ecological psychology. The concern of this author was how to account for the sophisticated adaptation of a living individual either animal or human, to its environment whatever the size of the brain of certain animals (sometimes very small) (Gibson 1979). In order to understand what an affordance means at the origin, one should both leave a perfect dichotomy between an individual and its surrounding environment and a symbolic vision of the processing of information.” (Morineau 2001, p. 83)

Besides the interaction of the individual with its environment, the affordance concept offers a non-cognitive approach to the stimuli provided by the environment. “A solicitation coming from a property of the environment and having an adaptive value for an individual is perceived in a straight way by it according to its biomechanics and sensory-motility characteristics. An affordance is first of all a perception which allows an immediate adaptation of the individual in the form of an action which takes into account this perception. The integration of the affordance in the perception-action loop does not need cognitive mediators implying signs the semantics of which would be stored within a declarative memory” (id., p. 84).

Ronald Stamper introduced affordance in the following way:

“Imagine the agent in a world of flux caused by the combination of his action and those in his environment. Within this ever changing world, experience teaches him the value of certain ranges of behaviour within which certain things are possible. The significance of each of these invariants is what it allows or does not allow the agent to do. In Gibson’s terminology each significant invariant represents a state of affairs or situation that affords or makes possible for the agent some repertoire of behaviour” … “from this point of view every object should be understood as a kind of conceptual shorthand for a repertoire of behaviour that it affords.” (Stamper 1997, p. 36)

We observe that in Stamper’s definition, the invariant is the value or the meaning of the range of behaviour.

According to Gibson, objects are perceptible by the means of perceptive actions that perform the subject. More precisely objects are identifiable thanks to the invariants of transformations caused by these perceptive actions.

Up to now we have shown how the conception of Gibson fitted the conception of action we have developed above.

Let us imagine now, the visual perception of a *piano* in Gibson's way. We will agree that it is not sufficient that an object looks like a piano to be a *piano*. One will then admit that it is necessary to use it to make music using its keyboard in order for it to be a *piano*. But making music is not just hitting keys. The keyboard must be used in such a way that we recognise it as music. But this way is not set by any particular individual but by a cultural context. And this cultural context is in no way produced by the acts of the individual when he is hitting the keyboard.

We will alternatively state that a *piano* is what permits all the actions that are allowed by the musical cultural norm. But again in this case there will not be a production of this norm thanks to the actions of the individual but to the execution of an action selected within a repertoire of actions called for in such circumstances.

We can sum up this situation in the form of a dilemma:

Either we retain the definition of an affordance given by Stamper after Gibson, and decide for instance that the object is indeed a *piano* for anyone sharing the same culture provided the object allows any action that is usually made with a piano. In this case we must admit that a selection of actions is necessary to achieve the cultural value and meaning of the object and consequently one is forced to abandon the notion of action we derived above. This is the case since it amounts to reintroducing even locally, the distinction between action and thought.

Or alternatively we want to retain strictly the concept of action we developed. In such an option with must renounce the fact that an object will be recognised with its usual cultural value and meaning, on the basis of all the action it affords to someone.

This problem has already been noticed by authors in the study of human–machine interfaces and more specifically in the design of ecological interfaces. However "the application of the concept of affordance to the domain of work has leads to a significant redefinition of the concept of affordance. The reason of this semantic evolution seems to be related to a deduction. In order to elaborate a model of the activity of an operator, it is necessary to account for the choices he makes among the complex set of information or affordances that he must face. This drives authors to raise the critical problem in the theory of Gibson of the selection of affordances among a set of prompting" (Morineau 2001, p. 84). From these criticisms several solutions were proposed (see Reed 1993; Vincent and Rasmunssen 1990). These solutions were based on a hierarchy of affordances.

"This organisation into a hierarchy then implies that a few strata of affordances typically considered as finalities constitute abstract elements that we can consider as possessing a symbolic representation on the cognitive level. On the other hand, according to the considered strata we obtain a dissociation between the rather internal affordances inside an individual (values, priorities, context and movement) and the rather external affordances (objects and environment). The affordance becomes therefore only indirectly the place of an interaction between the environment and the individual by means of a functional causality chain. These two dissociations go against … a strict definition of the notion of affordance: inscription of the individual among its environment and non symbolic cognitive control." (Morineau 2001, p. 84)

4.4 Affordances and Risks

In this section, we examine how risks can be described by affordances. We will point out that there are difficulties to achieve it. These difficulties are related to the sort of issues we have mentioned above.

First of all we now turn to risks as they are apprehended within technical domains.

What is a risk?

Risks are usually examined in safety analysis. Safety analysis is the activity the object of which is to identify, assess, reduce, accept, and control safety hazards and the associated safety risks in a systematic, proactive, complete, and cost-effective manner, taking into account the project's technical and programmatic constraints (Source can be found in ECSS 2003).

Safety analysis can be implemented through an iterative process, with iterations being determined by the project progress through the different project phases, and by changes to a given project baseline. Safety analysis comprises hazard analysis, safety risk assessment, and supporting analyses. Hazard analysis comprises the identification classification and reduction of hazards.

A definition of risk

According to Alain Desroches it is a "global concept of uncertainty as to the occurrence of a feared event, related to the likelihood of its occurring, the nature and the seriousness of its consequences but also to the perception that one might have of it" (Desroches 2004, pp. 53–57).

Dangers, dangerous situations, accident, and affordances

"Whichever the considered activity, human material or financial loss can be understood as the consequence of an accident which is itself the result of an accident scenario described by three sequential events: 'presence of a danger', 'dangerous or accidental situation', 'accident' (Fig. 1). These three events should be considered as states of the system and of its environment." (Desroches *et al.* 2003, p. 29)

"The *accident* corresponds to the 'realization' or the materialisation of the risk by human or material loss or the damages (material or immaterial)." (id., p. 30)

For example: a fire and an explosion of a truck in a tunnel.

"The *danger* or *threat* is the potential nuisance which could cause damage to people, goods, and to the environment. … Danger is the first link in an accident scenario. This one cannot exist in absence of a danger which is identified or not. Searching for one or several potential dangers during the running of an activity or during the mission of a system is fundamental. The result of this search will allow to intervene upon the conception of a system or upon the strategy of its exploitation. …

In the absence of danger it is not possible to identify events leading to dangerous situations." (ibid., p. 31)

For example: a truck running. But an inexperienced driver could a priori be considered as a danger independently of the vehicle he could drive.

"A *dangerous* or *threatening situation* is a state of the system in presence of a danger or a threat. The bringing nearer of the system and of the danger until their bringing together and their mutual covering is associated with the realisation of an event having either a random character or a deterministic one." (ibid., p. 32)

For example: a vehicle driven by an inexperienced driver could be considered as a dangerous situation.

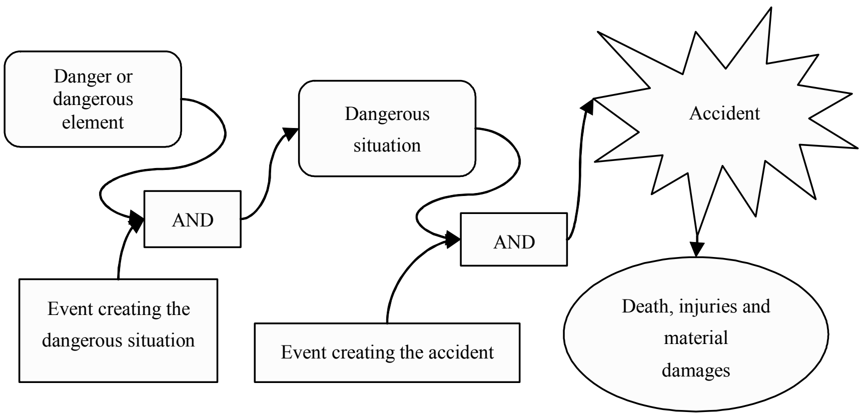

Fig. 1. A scenario of an accident can involve danger and a dangerous situation (Desroches *et al.* 2003)

From what precedes, it appears that a *danger* is perceived as such because there are *events* which could create a *dangerous situation* or could create an *accident*.

Therefore the element which corresponds to the danger, affords its dangerousness to the agent who brings about one of the two kinds of events which contribute to the accident. This sort of affordance is particular not only because it is "repulsive" and not "attractive", but because it is "accidental", that is it reveals a property of the corresponding element not belonging to the specific character of it, at least as it is usually considered or used.

But there is something more: because an accident involves a combination of factors as we have just explained, it is difficult for an agent usually concerned just by one of these factors to apprehend a danger and/or a dangerous situation. He should use a risk analysis in order to cope with the difficulty and carelessness due to familiarity of the affordance at hand. This issue is related to the problems we already mentioned and that were noticed by psycho-ergonomists who study the application of affordances in design of interfaces in case of virtual environment and of air traffic control (Morineau 2001). Instead of conceiving the activity of an operator as a procedure that should respect the instructions of the task, the affordance concept allows the conception of the activity as a possible space in which it is possible to navigate implementing operative strategies and learning. However there are issues about how to select the best affordance in order to achieve a task. One also should consider cases[2] where "a processing exclusively based upon affordances shortly lead to maladjustment of the subject: inadequate answer to a problem which needs an abstraction of reality. In order not to fall into a deadlock (which could be defined as 'local minima'), the person should possess an ability to inhibit the salient affordances in the environment …, to change his view on his environment (mental representation) and to activate knowledge related to his past experiences going beyond the adaptation to the immediate situation as it presents itself" (Morineau 2001, p. 88).

4.5 Risks in a Multi-Viewpoints Semiotics

Relating a semiotic approach to the concept of risk is based upon the hypothesis that the perception of a risk involves a human subject who grasps or produces signification; and in this case the elucidation of the

[2] These cases correspond to psychological tests inspired by Piaget (see Morineau, 2001)

conditions of this process can take the form of a semiotic theory. When expressed without preparation this hypothesis is a little surprising since the notion of risk is usually attached to what is unfamiliar to us, to what we do not control and which because of that seems to partly escape the domain of our subjectivity.

In several chapters (for presentation and references, see for instance Galarreta 2004), we have proposed a semiotics approach of technical systems which tries to conciliate the impersonal subjectivity – we called viewpoint – of an agent with the objectivity of collective designing and manufacturing in the form of space systems. We have proposed arguments in favour of a semiotic view of a risk: it is inseparable of an action that it characterises as a semiotic doing. Therefore risks are a priori good candidates to be apprehended by multi-viewpoint semiotics.

Description of a space system in a multi-viewpoints approach

In designing a technical system such as a space system, an issue is to find a common framework where the designers can efficiently share their knowledge of the same problem (see Galarreta 2004).

In a complex approach a technical system can be defined as the set of the views which comply with the set of (explicit) requirements which define the system on a functional plane and which also satisfy all the (explicit and implicit) physical constraints in order to assure a stable physical existence. We can extend this list to other requirements or constraints according to the viewpoints which are convoked in the production of these views.

We can give a more precise statement of this complex approach, by defining a viewpoint as the competency to produce or grasp the meaning of discourse and representations (contained in documents, schemas, images, etc.) in association to a trade. For instance we can distinguish viewpoints such as electrical viewpoint, mechanical viewpoint, thermal viewpoint, etc.

Instead of considering the space system designed by a team of designers from a single point of view (e.g. from a functional point of view or from an economical one) we proposed to consider the system just as a *signifying object*, the significance of which is to be a "space system" whichever the viewpoint we choose to observe it.[3] This means that the system is only virtual when it is observed from a single point of view (cf. Fig. 2). It is virtual and not actual, because it lacks all its other dimensions (=the other viewpoints). Only with all dimensions can it give an actual character to the system.

[3] During the design process this "observation" is either an interpretation or a production of a view.

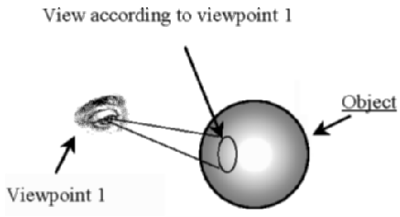

Fig. 2. A view of an object produced by a viewpoint

In the absence of interaction with other viewpoints the object is virtual and a view of it is a piece of data.

Within this theoretical framework, it is possible to give a precise definition to the concepts of *data*, *information*, and *knowledge*.

- A piece of *data* is a view with respect to a viewpoint of a (*virtual*) object.
- A piece of *information* is a view with respect to a viewpoint of an *actualised* object when a confrontation with other viewpoints occurs.
- A piece of *knowledge* is a view with respect to a viewpoint of a realised object as a result of a negotiation process with other viewpoints, assuming that a confrontation took place before.

The producing of a piece of knowledge therefore takes place during a negotiation process. This process is interpretable as the repairing of the *identity* (see Galarreta 2004), the identity of the object: (a) being designed or (b) manifesting an anomaly the cause of which is looked for, or (c) being the target of a risk analysis process. This negotiation process is based upon a *value system* (see Galarreta 2004).

Let us consider now the production of knowledge for its own sake in the three corresponding processes.

During a designing activity views of the object are virtual before the confrontation of viewpoints then they are actualised through confrontation and last realised when the negotiation process ends: they become a piece of knowledge.

In an analysis of the cause of an anomaly views are directly actualised through the confrontation of viewpoints involved in the description of the anomaly; they will evolve to become realised views as the repairing (negotiation) process is progressing. At the end they turn into a piece of knowledge.

Table 1 sums up these arguments:

Table 1. Table of comparison of the three negotiation processes

	The set of views and viewpoints involved is bound	The views and viewpoints involved are a priori known	Repairing of the identity
Designing	Yes	Yes	Occurs on the virtual views of the system
Analysis of cause of anomaly	Yes	No	Occurs on the actualised views of the system
Risk analysis	No	No	Occurs on the realised or actualised views of the system

In risk analysis, the views from which the analysis starts from, could correspond to realised views of the object (e.g. the system) before a possible confrontation is proposed through potential anomalies, their causes, and their effects.[4] When the risk analysis, preventing and protecting action have been proposed, the repairing of the object can be considered as complete and the views obtained correspond to a piece of knowledge.

Let us take an example proposed in Desroches *et al.* (2003). Let us consider a truck.

A manufacturer of trucks knows that a truck should simultaneously and permanently satisfy several requirements, i.e. views, in order to be both able and allowed to run on roads (Fig. 3).

When we consider a truck from the *viewpoint of road safety* it is usual to consider that a moving truck can represent a *danger*: it is often observed that drivers lose control of their truck with harmful consequences. It is usually because of their speed, of the surface of the road, of the shape of the trajectories that the road allows, the width of the road, and so on.

These dangerous conditions are evoked whenever we consider heavy vehicles such as trucks. They constitute *common places* of the *value system* of trucks manufacturers as well as of road safety authorities. They can challenge the "normal" identity of a (moving) truck. If one selects a situation among situations potentially encountered by moving trucks such as the entering of a tunnel, one can then select situations that can turn out to be

[4] Actualised views could be also confronted during risk analysis with potential anomalies if they occur during the designing activity and with temporary views.

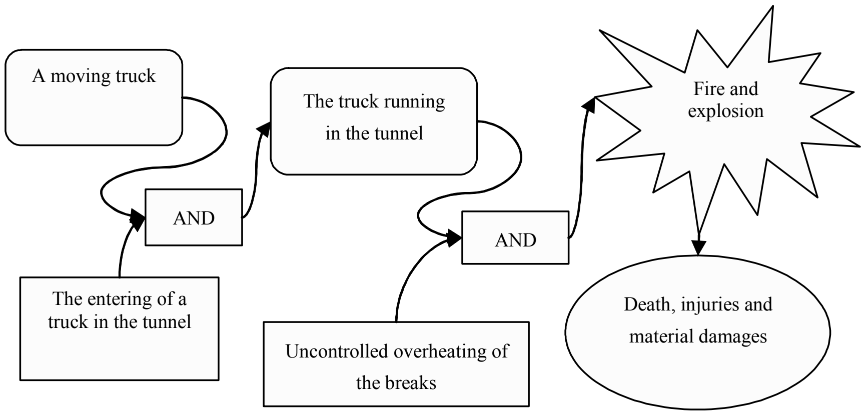

Fig. 3. Dangers and dangerous situations reveal the existence of common places. Example is taken from (Desroches *et al.* 2003)

dangerous. This selection again is based upon the knowledge of common places in the study of accidents. Again these situations may challenge the identity of a moving truck and consequently of a truck (if we admit that a truck should be able to move on a road).

We postulate (in Galarreta 2004) that a collection of viewpoints is structured by the existence of entities that manufacturers or designers for instance, could assimilate to products, but that we prefer to define as *theme* and *common places*:

- A *theme* is a potential place of correlation of viewpoints, which is established or verified through use; it is therefore an empirical concept.
- A *common place* is a realisation of such potential correlations. It can involve several themes.

A *system of values* in this context is the collection of viewpoints structured by the existence of themes and common places. The study of value systems within a multi-viewpoint semiotics needs to be carried out.

Risks experts strongly suggest that risks are related to perception. This position is congruent with the role that *perception* plays both in the implementation of action and in semiotics. “Perceiving something … is to perceive more ore less intensively a presence. … that is something which on one hand is situated in a certain location with respect to ours and has a certain extent, and on the other hand affects us with a certain intensity” (Fontanille 1998, p. 37).

Either perception of risks is synonymous with evaluation or: perception of risks is synonymous with focusing attention:

Perception of risks is synonymous with evaluation

"The first objective of risk management is to determine whether the identified risk is acceptable or not with respect to a predetermined scale. Depending on your past history and experience, if there is no scale, your perception of risk and decisions which have to be made to deal with it will be different. If you are involved in short term activity, you will probably pay more attention to the likelihood of the feared event occurring, a priori quite a weak criterion for determining risk which generally will mean nothing. On the other hand, if you intervene for example in a long-term project or activity, the likelihood of risk will be perceived as great and, while taking the seriousness into account, you will act differently when evaluating the risk. It then follows that you will do whatever is needed to reduce all or part of the risk." (Desroches 2004, p. 53)

Perception of risks is synonymous with focusing attention

People who work in risk management should doubt "Doubting means first of all, for each stage in a process, identifying elements which might prevent us from achieving it. We begin by looking for and consolidating related information which might slow down the operational approach but which will enable us on the other hand to avoid later problems. There has to be a balance between initial considerations and decision-making and corresponding action since doubt should not lead to sclerosis in decision-making and related action" (Desroches 2004, p. 53).

"You can't see a risk if you don't look for it. When it becomes 'visible', it is often too late to do anything about it. This is the case if we only treat risks by collecting facts which have occurred without paying enough attention to prevention. Everyone knows that any regulation, however perfect it may be at a given time, does not cover all risks and can even become inefficient in new, rapidly changing environments which go beyond its field of application." (Desroches 2004, p. 55)

These two acceptations of risk perception correspond to the general distinctions that semiotics make in perception and that we have just mentioned.

4.6 Managing Risks of Knowledge Evolution in a Long-Duration Space Mission

4.6.1 Rosetta: a long-duration space mission

The ROSETTA Mission of the European Space Agency (ESA) will study comet Churyumov Gerasimenko with which the probe has a rendezvous in August 2014.

After a period during which a global mapping of the comet will be realised by the orbiter, a closer observation phase will follow, including the sending of a module (Lander) down to the comet.

The launch that took place on 2 March 2004 by an Ariane 5 launcher, will lead to a placing in the right orbit near the comet by August 2014 for an 18-month observation period.

The International Rosetta Mission was approved in November 1993 by ESA's Science Programme Committee as the Planetary Cornerstone Mission in ESA's long-term space science programme. The mission goal was initially set for a rendezvous with comet 46 P/Wirtanen. After postponement of the initial launch a new target was set: Comet 67 P/Churyumov-Gerasimenko. On its 10-year journey to the comet, the spacecraft will hopefully pass by at least one asteroid.

In the case of a project such as Rosetta, and in addition to the risks that any project is faced with, the ESA's teams are also faced with the risk of losing critical knowledge when the Lander arrives on the comet after 10 years of space travelling.

4.6.2 Managing knowledge evolutions

The difficulty in detecting knowledge evolutions about an *object*, let us say for instance a *system*, is to identify which are the views which correspond to a piece of knowledge. If we admit the definition of a *piece of knowledge* we proposed above, we can notice that it refers to a dynamic process (confrontation/negotiation) which is difficult to detect when considering the static representation of it in a text, schema, or image, etc. We can only hope to discover stable structures which *are not* knowledge about the *object* but correspond to the culture in which the object is produced. This is what we defined above as *themes* and *common places* which characterise a *value system*. They are the semantic/semiotic elements which allow the negotiation/repairing process to take place and succeed.

A research study has been carried out in Centre National d'Études Spatiales (CNES) to tackle the question of knowledge preserving. Since it is out of reach to grasp the whole knowledge involved in the Rosetta Mission, we made proposals to maintain the capability of the organisation to produce a restitution of the risks attached to the system. This reduction of the knowledge domain to *already* identified risks is justified by the fact that these special pieces of knowledge address critical components of the Rosetta Lander (that we exclusively consider in this study).

The chosen strategy is a preventive one which consists of maintaining the ability of the organisation to detect sufficiently early the evolution of the value system of the considered mission (Condamines *et al.* 2003).

Characterisation of this values system is crucial in order to make explicit on one hand, the zones of the project where risk analysis has been thoroughly conducted[5] and on the other hand the zones where risks need not be treated.

As time passes, the values system is likely to evolve, and previous zones of risks may be forgotten.

Knowledge we have of a technical object are traces of the value system we live in. Some parts of this knowledge are related to the *nominal* aspects of the objects; other parts are related to *attached risks*.

Detecting evolutions among the information produces and used by a project is valuable only if it reveals knowledge evolutions or equivalently, evolutions of the value system. The traces of the evolution of the values system are to be found in

- Statistical distribution of terms within the project documents
- Impact on taxonomies of domains of the project
- Logical relations between "interesting" concepts
- Linguistic cues extracted from documents (see Condamines *et al.* 2003 for more details)

However results obtained by the research study carried out in CNES also indicated that a strategy based on text-mining techniques should be complemented by organisational solutions. For instance, it is not useful to automatically detect any (previously identified) risk if it turns out that certain risks are to disappear by the end of a given phase of the mission.

It should also be noted that the need to automatically detect previously identified risk leads to identify simple organisational solutions to the problem we consider. For instance, ask the expert which documents will be mandatory to know in order to restart a subsystem in 10 years.

[5] In those zones, quality insurance and quality control provisions have been foreseen.

4.7 Conclusion

In this chapter, we examined how a semiotic approach of risks can be proposed and how the concept of affordance can be fitted for such a goal. We are reminded that the primitive notion of action and its close relation with risks. The perception issue of the sensation was also examined through the work of the French philosopher Maurice Pradines and through the study of visual orientation by James J. Gibson. It turned out that on a perception plan their positions were similar. However the affordance concept does not belong to the primitive action paradigm.

A multi-viewpoints semiotics offers a convenient framework for defining a risk as a semiotic concept. We examined the question of managing risks in the special case of the Rosetta long-duration mission to prevent uncontrolled knowledge evolution. Managing risks of knowledge evolution should be based in this case on the combination of text-mining techniques and organisational arrangements.

These proposals are expected to be a contribution for a better semiotic approach of risk management.

References

Condamines A, Galarreta D, Perrussel L, Rebeyrole J, Rothenburger B, Viguier-Pla S (2003) Tools and Methods for Knowledge Evolution Measure in Space Project. Bremen, Proceedings of the 54th International Astronautic Congress, September 29 to October 3

Desroches A, Leroy A, Vallée F (2003) La gestion des risques. Collection Hermes Science, Lavoisier, Paris

Desroches A (2004) Before you manage a risk you first have to be able to doubt. Qualité Espace N 41, 06-2004, CNES, Paris, pp. 53–57

Dufrenne M (1987) L'œil et l'oreille, Editions de l'HEXAGONE, Montréal

ECSS (2003) Space product assurance. Hazard Analysis, ECSS-Q-40-02A, 14 February 2003

Fontanille J (1998) Sémiotique du discours, Presses Universitaires de Limoges, Limoges

Galarreta D (2004) Designing space systems in multi-viewpoints semiotics. In: Liu K (eds.), Kluwer Academic, Dordrecht, The Netherlands

Gibson JJ (1958) Visually controlled locomotion and visual orientation in animals. British Journal of Psychology, 49, pp. 182–194

Gibson JJ (1979) The ecological approach to visual perception, Lauwrence Erlbaum, London

Guendouz C (2003) La philosophie de la sensation de Maurice Pradines. Espace et genèse de l'esprit. coll. Europaea Memoria, I, 22. Hildesheim_Georg Olms Verlag

Lalande A (1985) Vocabulaire technique et critique de la philosophie, Presses Universitaires de France, Paris

Morineau T (2001) Eléments pour une modélisation du concept d'affordance In Congrès PIQUE de la Société française de psychologie. Nantes septembre 2001, pp. 83–95

Pradines M (1909) Principes de toute philosophie de l'action, Félix Alcan, Editeur, Paris

Reed ES (1993) The intention to use a specific affordance: a conceptual framework for psychology. In: Wosniak RH and Fisher K (eds.), Development in Context, Acting and Thinking in Specific Environments, Lawrence Erlbaum, Hillsdale, NJ, pp. 45–75

Saint-Sernin B (1989) Génèse et unité de l'action, Librairie philosophique J. Vrin, Paris

Stamper R (1992) Signs, organisation, norms and information systems. Proceedings of the 3rd Australian Conference Information Systems, ISOP-92 Wollongong, NSW, Australia, 5–8 October

Vincent KJ, Rasmunsen J (1990) The ecology of human-machine systems II: mediating "direct perception" in complex work domains. Ecological Psychology, 2, pp. 207–249

Part 3
Organizational Semiotics and Multi-Agent Paradigm

Chapter 5
Norm-Based Contract Net Protocol for Coordination in Multi-Agent Systems

J. Wu, R. Gan

School of Management and Economics, Beijing Institute of Technology, Beijing 100081, China

Abstract

Contract net protocol (CNP) is often used for coordination in a multi-agent system (MAS). Due to the limitations inherent in the conventional CNP, this chapter proposes a norm-based CNP to improve the efficiency and effectiveness of the coordination processes in a MAS. Firstly, a three-dimensional taxonomy of norms is put forward in terms of the hierarchy, type, and flexibility of norms. Then a coordination process guided by norm-based CNP is developed under the taxonomy framework. The new coordination process consists of two principal stages and five subdivided phases, providing a feasible solution for the optimization of the candidate selection. A case study is finally presented to illustrate the real application of the proposed approach.

5.1 Introduction

Multi-agent systems (MASs) have caught more and more attention in recent years. These systems have been applied in a variety of domains such as manufacturing, electronic commerce, and traffic control. In MAS, an agent is an entity that is situated in some environment and capable of acting autonomously in order to meet its design objectives (Wooldridge 2002). An agent is used to denote a human or computer system software with the

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 91–107.

following properties: bounded autonomy, rationality, social ability, reactivity, and responsibility.

A crucial problem in MAS is the balance between autonomy of individual agents and coordination required between agents to complete efficiently complex tasks. In MAS, one agent cannot solve the complex problem solely since it has no sufficient competence, resources, or information. Agents will influence others to convince them to act in a certain way. Even if agents have common interests to cooperate, they may still have conflicting interests to be coordinated. So cooperation and coordination are critical for managing such inter-agent dependencies and reaching global optimization.

Coordination is a means for members in a system to communicate and compromise to reach mutually beneficial agreements regarding belief, goal, or plan. Coordination is a kind of dynamic glue that binds tasks together into larger meaningful wholes. The complex coordination is achieved by structuring mutually constrained entities into a whole, integrated and harmonious adjustment of individual work efforts towards the accomplishment of a larger goal (Ossowski, 1999). By such process an agent reasons about its local actions and the foreseen actions that other agents may perform, with the aim of making the community behave in a coherent manner. Each agent performs its own tasks and completes jobs for the whole system through communication and coordination with other agents.

The protocols, objectives of coordination, and behaviour mechanism of the agent are essential for coordination. Coordination consists of a set of mechanisms necessary for the effective operation of the MAS in order to get a well-balanced task division while logical coupling and resource dependencies of the agents are reduced.

This work is concerned with the protocol issues for coordination in MASs. Section 2 introduces the Contract Net Protocol (CNP) which is widely used for coordination among agents in MASs. The inherent limitations of conventional CNP are analysed. A taxonomy of norms for the proposed norm-based CNP is studied in Section 3. Section 4 is devoted to the essential procedures of norm-based CNP in coordination process of MAS. A case study is presented to illustrate the application of the proposed approach.

5.2 Contract Net Protocol

A protocol is a set of rules agreed among the members of the system for their interactions and communication. The CNP provided by Smith and Davis is often used for coordination among the nodes in a network system during problem solving (Smith 1980; Smith and Davis 1981). A node can be either a manager (contractor) who monitors the task's execution and processes the results of its execution or a contractor, who is responsible for the actual execution of the task. The manager and contractor are only roles and any node can take on either role dynamically during the course of problem solving. The contract between a certain manager and contractor is established by the process of mutual selection based on a two-way transfer of information. The process may include the following activities. Firstly, the manager makes an announcement of the task required. Then potential contractors evaluate the tasks announced and submit bids for the tasks they are interested in. Finally the managers evaluate the bids and award contracts to the winners of the bidding. The contract is an explicit agreement between a manager and a contractor.

The CNP offers a formal procedure in the coordination process in a network system and is widely employed in MASs (Xu and Weigand 2001). However, conventional CNP just offers rough steps and briefly sketched processes. Since it lacks specified criteria for the managers and contractors to obey in the coordination process, several complications arise in its implementation, and the knowledge and rules which exist in the coordination processes are often overlooked. Furthermore, broad announcement and acceptance of the messages will cause message overflow and redundancy problems. This might lead to the saturation of the capacity of related agents in the MAS. In view of that mentioned above, therefore, the coordination processes in the network system may not normally be controlled and the efficiency and effectiveness of the coordination processes may not meet the objectives and goals of the MAS by conventional CNP.

5.3 A Taxonomy of Norms for Norm-Based Contract Net Protocol

This work is devoted to the extension of CNP employed in the MAS to deal with the issues mentioned above. A framework of norm-based CNP referring to the concepts and methods of organizational semiotics is proposed in order to provide a normal and clear coordination procedure in the system. Norms in the process act as basic guidelines for interaction between agents. A rational and clear classification of norms in coordination processes with norm-based CNP will greatly increase the efficiency and effectiveness of interaction and reasoning processes and reduce the consumption of resources.

Wright (Wright 1963) explains the concept of a norm in this way: “Norm” has several particular synonyms which are good English. “Pattern”, “standard”, “type” are such words, so are “regulation”, “rule”, and “law” (Liu 2000). And according to sociology, a norm is a rule or standard of behaviour shared by members of a group. Norm is developed through practical experiences of people in a culture, and in turn has functions of directing, coordinating, and controlling actions within the culture. It provides guidance for actors to determine whether certain patterns of behaviour are legal and correct within the given context.

In norm-based CNP, norms are layered and classified to normalize the reasoning process of agents and satisfy various needs in order to improve the effectiveness and efficiency of the coordination processes. MAS can be seen as a social system where institutional and organizational regulations as well as business and operational rules are defined as norms and shared among agents. In this connection, norms are hierarchically divided into three layers from higher to lower as: social norms, organizational norms, and operational norms. The lower layers should obey the higher layers and higher layers have priority over lower ones, illustrated briefly as follows:

- *Social norms* include cultural, religious, ethical, moral, and legal regulations as well as social conventions (Briggs and Cook 1995; Vázquez-Salceda 2004). For example, consider a domain in which enterprise organization must obey environment laws.
- *Organizational norms* include organizational goals, objectives, regulations, and culture. For example, the sales volume of a company is planned to increase by 10% more than last year.
- *Operational norms* include business rules, administrative and technical standards, operational regulations, and procedures. For example, if a customer did not pay for a previous order and the sum of the order was less than ¥100,000 he must pay the bill before the delivery of the products.

Norms can also generally be divided into two classes: rigid and flexible classes.

Rigid class
Norms in rigid class must be obeyed by the agents. For example everyone should comply with the dictates of the law without having a choice.

Rigid norms existing in MAS are viewed as global constraints that a group of agents must satisfy to ensure the successful achievement of a system’s goals.

Flexible class

Norms in the flexible class give the agents the authority to decide what to do in a certain situation. For example, the sales manager has the authority to negotiate with the customer to determine delivery dates of orders placed.

Flexible norms provide the flexibility which cooperating agents need if they are to cope within dynamic environments.

According to the ways of controlling human behaviour, norms are divided into five categories: perceptual norms, cognitive norms, evaluative norms, behaviour norms, and denotative norms. These are further elaborated in Stamper *et al.* (2000). The perceptual norms deal with how people receive signals from the environment via their senses through media such as light, sound, and taste. The cognitive norms enable one to incorporate the beliefs and knowledge of a culture, to interpret what is perceived, and to gain an understanding based on existing knowledge. The evaluative norms help explain why people have certain beliefs, values, and objectives. The behavioural norms govern people's behaviour within regular patterns. Finally the denotative norms direct the choices of signs for such signifying choices and are culture-dependent, e.g. the choice of a colour to signify happiness or sadness.

With this connection, a three-dimensional taxonomy of norms in the norm-based CNP is proposed and illustrated in Fig. 1.

Norms guide the processes of coordination and cooperation. Though norms may limit the autonomy of agents, they ensure that the system's goals can be achieved as a whole and reduce the cost of interactions among agents.

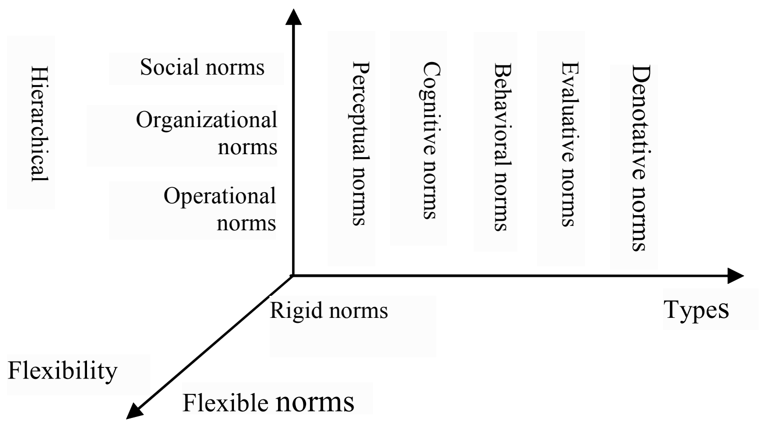

Fig. 1. A three-dimensional taxonomy of norms

5.4 Coordination with Norm-Based CNP

The coordination process guided by norm-based CNP is composed of five phases as follows:

Phase 1: Task announcement
This phase is devoted to the announcement of tasks. Managers prepare the announcements of the tasks and issue them to every agent. Table 1 shows the template of a task announcement which is composed of task name, task content description, eligibility specification, and expiration time. Eligibility specification indicates the obligatory requirements for which only suitable agents are allowed to bid. Expiration time is the deadline for accepting the bid. Table 2 shows the main content of a bid which consists of bidder name, bidding task description, and bidding constraint and expiration time.

Phase 2: Evaluation of the task and bids based on rigid norms
Step1: Task evaluation by receivers of the announcement

Eligibility specification and expiration time as a form of rigid norm restricts the number of receivers who bid. It ensures that the bidders are capable of meeting the obligatory requirements for bidding and they are interested in the announced tasks. The receivers rank the suitable and attractive tasks and decide to bid or not, according to their interests. If the bidder is not awarded the task before expiration time the bid may be eliminated. This way the bidder has the right to sign contracts with other managers.

Step 2: Evaluation of bidders by managers

In this phase, rigid norms should be given attention. This helps to reduce extraneous message transmission and speed up bids processing. For example, the expiration time eliminates the agents who do not respond on time automatically; also the overdue bids are annulled, which avoids the invalid negotiation process.

Table 1. List of main contents for bid request

Task name
Task content description
Eligibility specification
Expiration time

Table 2. List of main contents for bid

Bidder
Bidding task
Bidding constraint
Expiration time

The bidder evaluation by managers include, credit information and the capacity of the bidders to identify the eligible bidders.

Credit information of the agents plays a significant role in the coordination process in MAS. In the bid evaluation process, credit information for each related agent should be checked. Agent's credit information is recorded in the information server of the system.

The capacity of each bidder is then evaluated. For instance, if the bid is for a manufacturing task, the key factors of evaluation may involve financial status, production capacity, technical ability, management and organizational situation, historical records of the performance and so forth.

As a result of evaluation in this phase, a few eligible bidders are selected as candidates for contracting and/or for further negotiation. The candidates will receive the awarding messages from the managers.

Phase 3: Negotiation between the manager and candidates based on flexible norms

Negotiation is a process by which a joint decision is made by two or more parties through a process of concession or a search for new alternatives. Negotiation is a coordination mechanism and one of the ways to achieve coordination, which is based on commitments within a group of agents. Negotiation processes dynamically generate agreements which usually last for less time than the prior commitments that organizations imply.

Agents are in different domains and have various interests, beliefs, and knowledge. Final agreement is achieved by mutual selection through negotiation.

The specific development of the negotiation process in a particular situation is determined by the negotiation strategies of the parties involved. These strategies determine which of the permitted options an agent actually chooses. They are purely local decision criteria, which need not be constrained by any external convention. In negotiation each party involved in the contract evaluates information from its own perspective and compromises according to flexible norms. Usually an agent's negotiation strategy in decision-making aims at the maximal satisfaction of its self-interest. Utility functions act as evaluation strategy in decision-making.

Phase 4: Contract awarding

Successful negotiation leads to a contract. A contract is established by a process of mutual selection based on a two-way transfer of information. When a contract is signed by manager and contractor, they have a commitment to accomplish it. The information of the signed contracts is recorded in the information server of the system.

Phase 5: Task execution

Once a contract is signed, the parties involved start to execute the task according to their commitments. When the contract is terminated and the task is accomplished, performance evaluation should be carried out with related norms. Perfect performance of an agent will lead to the improvement of its credit level and vice versa. In the case where an agent violates its commitment during the task execution, it should be penalized according to the norms of punishment criteria.

The above phases in coordination process may be grouped into two stages, namely the stage of preliminary evaluation which consists of phases 1 and 2 and the stage of final decision-making which consists of phases 3, 4, and 5. The coordination processes with norm-based CNP is illustrated in Fig. 2. After the stage of preliminary evaluation, the number of agents to be negotiated may be greatly reduced. It will lead to the improvement of the efficiency and effectiveness of the coordination process in MAS.

One of the merits of dividing the process into two stages is the convergence of coordination process as soon as possible and the control of the whole system effectively. In coordination process, norms ensure to achieve global performance and agents pursuing maximal local performance to achieve a satisfactory result.

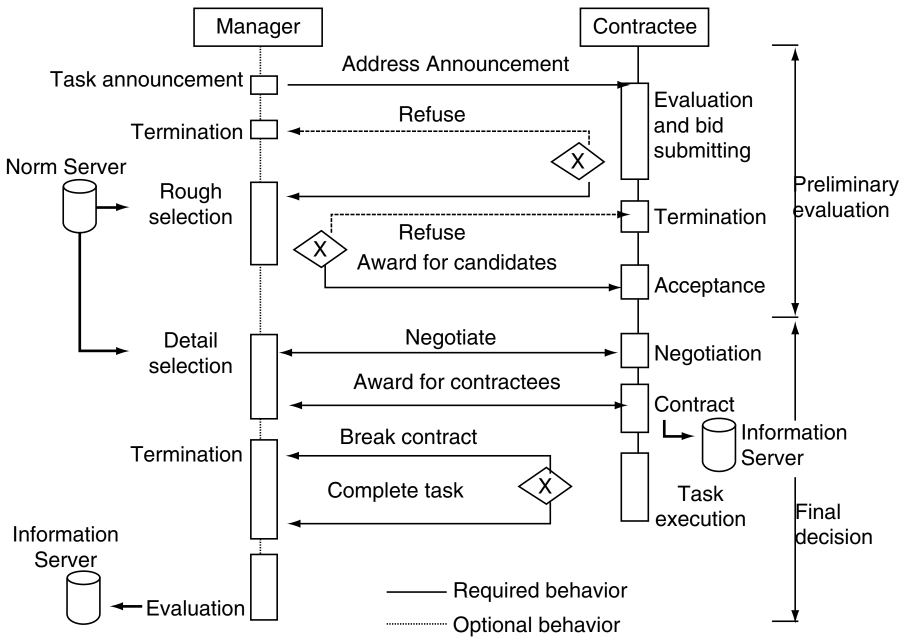

Fig. 2. Interaction chart with the norm-based contract net protocol

5.5 A Case Study

5.5.1 Agent modelling of the enterprise

The target enterprise for the case study is a large-scale jewellery manufacturing company in China. We indicate the name of the company in the case study as CHJ Co. During past years, the production competence of the enterprise was relatively poor contrary to the booming market. The main reasons are that CHJ Co. could not react quickly and schedule a manufacturing process when receiving urgent orders. CHJ Co. was faced with the rigours of business. For example, over 20% of the orders from chain stores had been postponed and even cancelled since 2003. The board of the company decided to reform the management systems to improve the agility, flexibility, and efficiency of the enterprise. One of the main measures was to promote and normalize the negotiation and coordination among various departments and divisions of the company with the support of information systems. Multi-agent technology is used to model and simulate the negotiation and coordination mechanism and deal with the dynamic and flexible scheduling, and particularly, to handle unforeseen circumstances.

Agents are the basic entities in our framework. Each workshop and cell is encapsulated as an agent. Each agent, having equality and a certain authority in this enterprise, competes and cooperates for dynamic resource allocation. The planning agent assigns tasks according to customer's order and traces its execution. The purchasing agent takes charge of purchasing material on the basis of material repertory and production requirements. The sales agent is responsible for the order management and the customer relation management. Workshops have various types of machines and staff to yield diverse jewellery.

The businesses of the entire enterprise are accomplished under the coordination and cooperation of agents. Though the agents finish various tasks driven by their own local data, some of them should abandon local strategies if needed for the global goal.

The enterprise's concept model with agents is illustrated in Fig. 3.

The conceptual model of an agent is shown in Fig. 4. Each agent should at least have knowledge of the capability, availability, and cost of the physical resource (e.g. a machine) that it represents. Each agent consists of the following components: identity information, communication module, negotiation module, resource module, ontology database, norm database, decision module, and executing module.

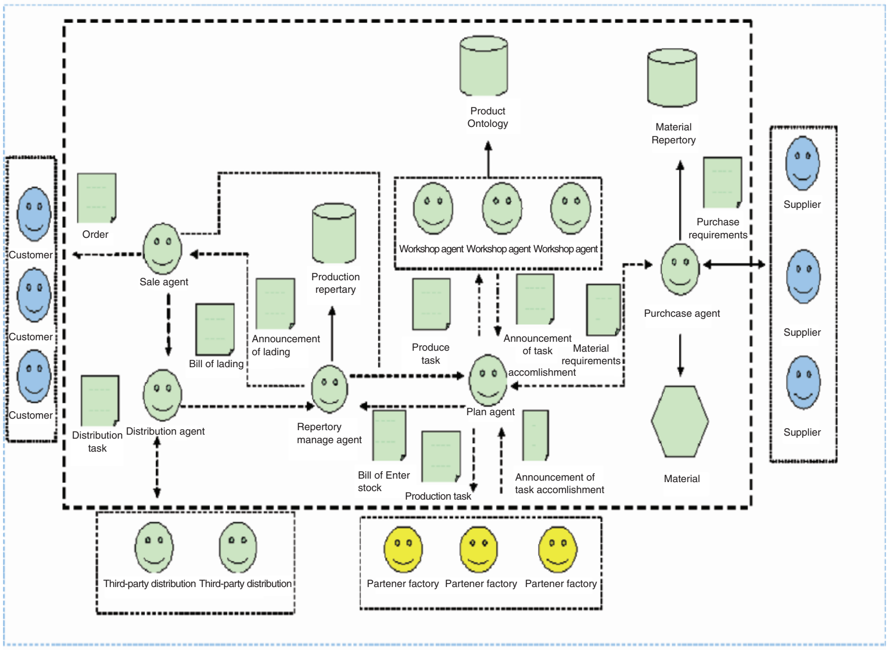

Fig. 3. An agent-based model of the company

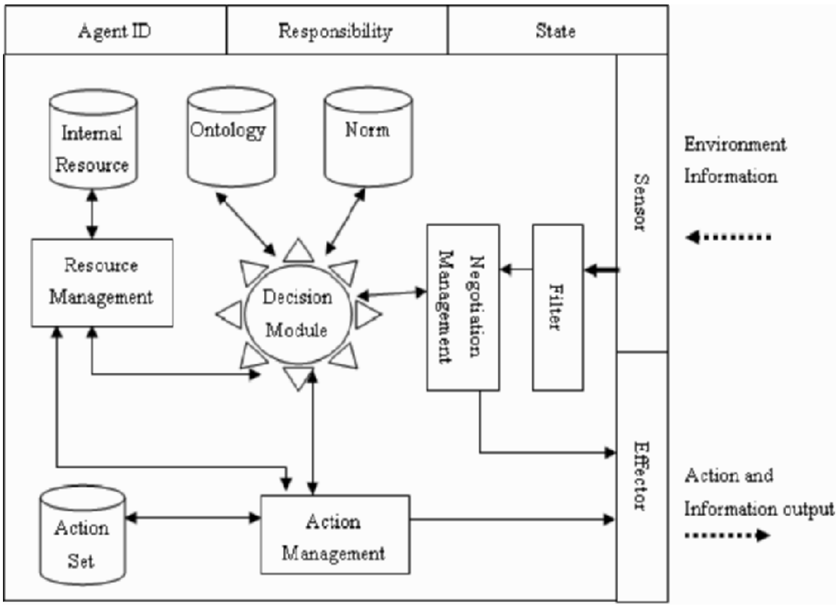

Fig. 4. Conceptual model of the agent

Identity information includes agent name, responsibility, and state. Communication module, including sensor, effector, and filler, is in charge of collecting information and transmitting it to relevant modules. The resource module is responsible for obtaining, supplying, and reclaiming resources. The negotiation module manages agent's negotiation action and monitors agent's action in the light of the negotiation protocols. The Agent ontology database stores knowledge of individual agents while the enterprise ontology database stores common knowledge. The merits of differentiating enterprise and agent ontology into different levels are sharing, reusing data, and supporting agent reasoning.

Norm database stores individual agent's norms. In the same condition, different agents will have dissimilar reactions according to their own available knowledge and norms. The decision module is a core and command centre, which directly or indirectly controls the other modules. Decision module guides agent behaviours and sets down the task scheme and reasons considering the environment information, knowledge, and norms. The executing module manages action sets and performs task process according to task process and task schemes.

5.5.2 Application of norm-based CNP to the MAS

To survive in this competitive market, CHJ Co. must increase their productivity and profitability through agility. Multi-agent-based coordination may help to dynamically and flexibly schedule manufacturing processes and rapidly respond to market demands. In this work, a norm-based CNP is applied to this multi-agent-based enterprise information system. The negotiation processes and protocols among the agents are described as follows.

When CHJ Co. receives an urgent order, the enterprise should react quickly. Since the throughput of some divisions in the enterprise may be saturated, and each workshop makes independent accounting, the plan agent should coordinate with other agents to carry out scheduling for the new order. The allocation of the emergent orders must obey a set of norms or constraints that reflect the temporal relationships between manufacturing activities and the capacity limitations of a set of shared resources. Further negotiation between agents is also needed.

The allocation also affects the optimization of the schedule with respect to the criteria such as cost, time, quality, and throughput capacities. In terms of the particular conditions of an enterprise, we choose time, quality, and cost as the key factor to write an expected utility function as follows:

$$U = W_1 * Q + W_2 * C + W_3 * T \quad (1)$$

W_1, W_2, and W_3 denote the weights of quality, cost, and time respectively. Weights can be assigned by expert evaluators in consideration of the enterprise manager's interests and industry characteristics. Weight assigned has the properties such as:

$$w_i \in [0,1]\,;\ \sum_i w_i = 1\,;\ i \in (1,2,3) \tag{2}$$

Since the situation the enterprise faces changes dramatically, the weight should be dynamically assigned according to flexible norms:

Whenever <receiving the order > *If* < plan agent is time-preferred > *Then* <plan agent> *Oblige to* <assign W_3 = Limited (0.5, 0.8)>. (As norm number #N3)

Whenever <receiving the order > *If* < plan agent is quality-preferred > *Then* <plan agent> *Oblige to* <assign W_1 = Limited (0.5, 0.8)>. (As norm number #N4)

Whenever <receiving the order > *If* < plan agent is profit-preferred > *Then* <plan agent> *Oblige to* <assign W_2 = Limited (0.5, 0.8)>. (As norm number #N5)

The plan agent has authority to choose weight in the light of these flexible norms.

Specially, in the norm-based CNP, some social norms should be given more attention. For example, staff working hours should be within 8 hours per day according to labor laws. To avoid violating the labor laws, the enterprise should decompose and analyse the workshop agent's bid. The norm is shown as: Whenever <receiving workshop agent's bid > If < staff working-hours>8 hours> Then <plan agent> Oblige to <reject the bidder>. (As norm number #N6)

After an agent accomplishes tasks, the plan agent will evaluate its performance using quality evaluation norms. An example of a quality evaluation norm is that: whenever <quality checking> if <proportion of first class product not exceeds 90% > and <proportion of second class not less than 10%> and< proportion of the faulty not below 1%> Then <workshop agent> Oblige to <be punished>. (As norm number #N7)

If the agent cannot accomplish the task he has caused a delay, it should be penalized. An example of norms about penalty is that:

Whenever <break the contract> if <the agent is an enterprise workshop agent> then <workshop agent> oblige to <pay a fine of 10% of the contract payments >. (As norm number #N8)

In this example, the norms #N1 and #N2 are regarding eligibility specification and expiration time, respectively.

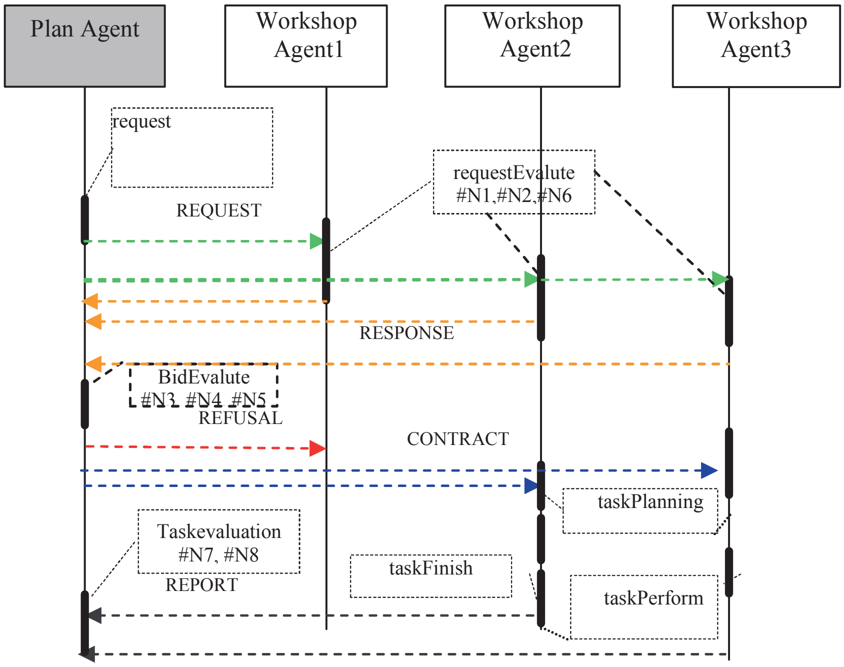

Fig. 5. The interaction view of multi-agent coordination

Figure 5 shows the whole coordination process.

Finally, a simulation has been constructed to validate the models obtained by the methods proposed using Swarm software. Some indexes have been designed to show the difference between using and not using norm-based CNP.

We illustrate only the main indexes such as operation efficiency and order acceptance.

$$OE = \frac{1}{N} \sum_{i \to N} \frac{PT}{ET} \tag{3}$$

OE: Operation efficiency
N: Total Number of Order
PT: Order Planned Time
ET: Order Execution Time

The statistics of operation efficiency show the capability of effectively executing order. The higher value of OE indicates the efficiency of the company is higher and the cooperation between agents is better.

$$OA = \frac{AN}{RN} \tag{4}$$

OA: Order Acceptance
AN: Number of Acceptance Order
RN: Number of Received Order

The statistics of order acceptance show the condition of order acceptance vs. received order with the growth of business in period of time. The higher value is the stronger competence of the company.

We simulate the behaviour of agents' coordination at receiving an order and allocating tasks. In simulation, data is randomly created referencing the company's historic data. The number of agents is designed to 50 which can be readjusted.

Pictures simulated of the MAS with Swarm have been shown as follows. Comparing Figs. 6 and 7, it is easy to recognize that the operation efficiency is improved using norm-based CNP. In contrast with the Fig. 8, Fig. 9 shows the order acceptance is obviously advanced by norm-based CNP. Table 3 shows the recorded data of the indices from the simulation process while the process becomes stationary.

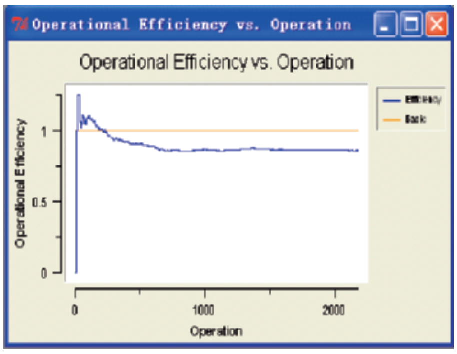

Fig. 6. Operational efficiency without norm-based CNP

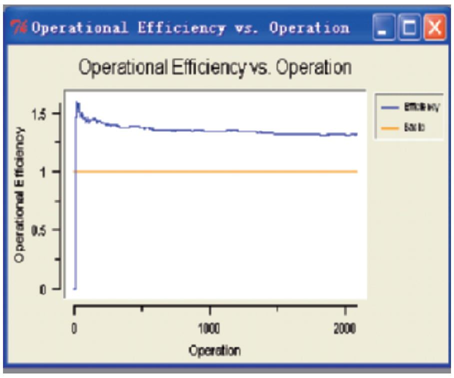

Fig. 7. Operational efficiency with norm-based CNP

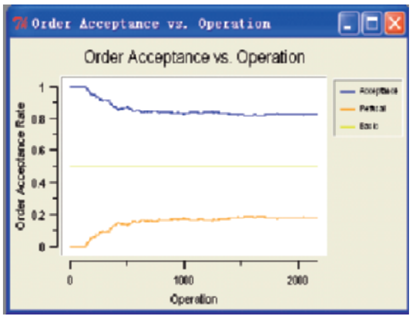

Fig. 8. Order acceptance without norm-based CNP

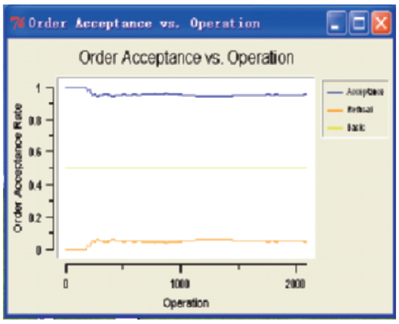

Fig. 9. Order acceptance with norm-based CNP

Table 3. Data of indexes OE and OA from the stationary simulation process

Item	Without norm-based CNP	With norm-based CNP
Operational efficiency (OE) index	[0.8622534639517689]	[1.3168070180202414]
	[0.8623389007529491]	[1.3161578233111835]
	[0.8623389007529491]	[1.3142331651858028]
	[0.8626291477261501]	[1.3142331651858028]
	[0.8626291477261501]	[1.3142331651858028]
	[0.8626291477261501]	[1.3142331651858028]
	[0.8626291477261501]	[1.3157966681602833]
	[0.8629446408164864]	[1.3157966681602833]
	[0.8621833606841957]	[1.3157966681602833]
	[0.8621833606841957]	[1.3157966681602833]
Order acceptance (OA) index	[0.905123339658444]	[0.9517543859649122]
	[0.905123339658444]	[0.9518599562363238]
	[0.9052132701421801]	[0.9518599562363238]
	[0.9052132701421801]	[0.9518599562363238]
	[0.9053030303030303]	[0.9518599562363238]
	[0.9053030303030303]	[0.9519650655021834]
	[0.9053926206244087]	[0.9519650655021834]
	[0.9045368620037807]	[0.9519650655021834]
	[0.9045368620037807]	[0.9519650655021834]
	[0.9046270066100094]	[0.9520697167755992]

5.6 Conclusions

Coordination is essential for MAS to achieve complex tasks and reach global optimization. Conventional CNP offers a formal procedure in the coordination process for a distributed network system, but it lacks specified criteria for both the managers and contractors to obey in the coordination process and this may lead to the inefficiency and ineffectiveness of coordination. A norm-based CNP is proposed in this chapter to provide normal and clear procedures of coordination among cooperative agents, addressing the problems born by conventional CNP. In order to normalize the reasoning process in coordination, a three-dimensional taxonomy of norms is studied. Norms are categorized into three layers, namely social, organizational, and operational norms, and two classes, namely rigid and flexible class, together with five varieties of norms: perceptual, cognitive, evaluative, behaviour, and denotative. A five-phase procedure of coordination is developed for the norm-based CNP and applied to a real case. The case study shows that the five-phase coordination procedure, generalized by two candidate selection stages, improves the effectiveness and efficiency of coordination by introducing the norm-based CNP.

References

Briggs W, Cook D (1995) Flexible Social Law, Proceedings of the International Jonit Conference on Artificial Intelligence (IJCI-95), Morgan Kaufmann, pp. 688–693

Liu K (2000) Semiotics in Information Systems Engineering. Cambridge University Press, Cambridge

Ossowski S (1999) Co-ordination in Artificial Agent Societies: Social Structures and Its Implications for Autonomous Problem Solving Agents. Springer, Berlin

Smith RG (1980) The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Transactions on Computers, C-29(12), pp. 1104–1113.

Smith RG, Davis R (1981) Frameworks for cooperation in distributed problem. IEEE Transactions on Systems, Man, and Cybernetics, SMC-11(1), pp. 24–33

Stamper R, Liu K, Hafkamp M, Ades Y (2000) Understanding the roles of signs and norms in organizations – a semiotic approach to information systems design. Behaviour and Information Technology, 19(1), pp. 15–27

Vázquez-Salceda J (2004) The Role of Norms and Electronic Institutions in Multi-agent Systems, Birkhäuser Verlag, Basel, Switzerland

von Wright GH (1963) Norm and Action: a logical inquiry. Routledge. London, 1963

Wooldridge M (2002) Intelligent agents: the key concepts. In: MASA 2001, LNAI 2322, Springer, Berlin, Heidelberg, pp. 3–43

Xu L, Weigand H (2001) The evolution of the contract net protocol. In: WAIM 2001, LNCS 2118, Springer, Berlin, Heidelberg, pp. 257–264

Chapter 6
Interaction of Simulated Actors with the Environment

H.W.M. Gazendam[1]

Groningen University/Twente University, P.O. Box 800, NL-9700-AV Groningen, The Netherlands

Abstract

The current chapter explores the possibilities of an improvement of the interaction of an ACT-R actor with its environment including other actors. This is done in the framework of a project aiming at a multi-actor simulation environment based on the ACT-R architecture. Two objections against traditional cognitive architectures like Soar and ACT-R, namely the lack of physical grounding and the lack of symbol grounding, are explained. For a possible improvement of this situation, organizational semiotics and simulation of emotion seem to offer promising perspectives. Organizational semiotics offers us concepts for the encoding of the environment in the form of affordance signs, social constructs, and social norms. This leads to new declarative chunk types in ACT-R. An emotion subsystem can maintain an emotional state that encourages task performance, learning, and social behaviour. An awareness subsystem enables task switching based on the emotional state and the selection of those social constructs and norms that are applicable to the current situation.

[1] The author wishes to thank Martin Helmhout for his comments on this chapter.

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 109–130.

6.1 Introduction

We aim at the development of a multi-actor simulation environment based on ACT-R that can be used for experimentation with concepts from organizational semiotics and organization theory (Roest 2004; Helmhout *et al.* 2004, 2005a,b). However, at the beginning of our project, ACT-R knew only a single-actor implementation that could not be used for multi-actor simulations. Our first task, therefore, has been to make ACT-R suitable for multi-actor simulation.

Thus far, we have succeeded in rebuilding ACT-R as a Java program, and have realized several enhancements that enable actors to interact in a multi-actor environment (Roest 2004; Helmhout *et al.* 2004, 2005a,b). The single-actor ACT-R program has been replaced by a client-server system in which each actor is a client running on some machine, and the simulated physical environment is the server running on some (maybe other) machine. The communication between actors (and environment) is based on TCP/IP sockets, the FIPA protocol, and XML messages. The ACT-R memory organization has been made more flexible in order to enable new memory access structures (buffers) that are necessary for handling interactive behaviour like movement, perception, communication, and social behaviour. A module for perception and movement in two-dimensional space has been added, as well as a module for the sending and interpretation of (XML) messages. For running experiments, a module for the collection of data and storing these in an external database has been realized. The user interface enables running simulations, and inspecting the contents of each actor's memory contents in terms of declarative chunks and productions (and their activations). Experiments with actor interaction while moving in a two-dimensional world have been done, showing the formation of (tacit) social constructs (Helmhout *et al.* 2005b).

This chapter explores the possibilities of getting a further step ahead in the direction of an improvement of the interaction of the ACT-R actor with the environment and with other actors. For possible improvements we have turned to two areas: organizational semiotics and simulation of emotion. Organizational semiotics offers us a concept for the encoding of signals from the environment and the actor body in the form of affordances, and gives us the concepts of social construct and social norm as regulators of socially acceptable behaviour. Simulations of emotion give us the emotional state that, when monitored, could be used for flexible switching between goals and for the encouragement of social behaviour.

6.2 The Use of ACT-R in a Multi-Actor Simulation

6.2.1 ACT-R and Soar as cognitive architectures

Actors in a multi-actor simulation can display intelligent behaviour if they are based on cognitive architectures like Soar (Newell 1990) or ACT-R (Anderson 1990; Anderson and Lebiere 1998; Lebiere, 2002). Soar and ACT-R provide the best simulation models of human intelligence known thus far with respect to higher cognitive functions like language use and solving problems.

Soar and ACT-R both are goal-*oriented production systems*. A production system consists of a set of production rules (or, for short, productions), and a set of data structures representing the state of the system (Newell 1973). Each production consists of a condition side and an action side. In goal-oriented systems, the existence of a goal of a certain type is part of the condition side. The production system works by selecting the productions that match with the current goal and data structures, and firing the selected productions. Firing means executing the action side of a production; this results in a change of goal structures and data structures. Based on the changed goal structures and data structures, a new set of productions is selected and fired. This goes on until no production can be selected, for instance because the goal collection is empty.

Soar and ACT-R differ in a number of ways.[2] ACT-R has a long-term *declarative memory* separated form the long-term procedural memory. In the declarative memory, there are declarative chunks, while in the production memory there are productions. Declarative memory is very flexible, while proceduralized knowledge has abandoned flexibility in favour of efficiency in access.[3] Soar does not have a long-term declarative memory, which means that Soar can only learn declarative facts by a very complicated mechanism called data chunking.

In some production systems, it is possible to match many production rules and fire these in *parallel*. This means that an unbounded amount of computation could be done in one production cycle. In a simulation of human cognition, this is not very plausible. Therefore, ACT-R allows parallelism in the matching of productions to the current goal, but only allows the production with the highest expected utility to fire. Although Soar allows for the matching and firing of many productions in parallel, this is only done for collecting information needed to select the operator to be applied. After that, only one operator is selected which is more or less

[2] Anderson and Lebiere, 1998, p. 439.
[3] Anderson and Lebiere, 1998, p. 29.

equivalent to allowing only one production to fire and change the goal state and other data structures. A major difference between Soar and ACT-R is that Soar places conflict resolution (the selection of the goal to fire) at the symbolic level, while ACT-R handles this at the subsymbolic level.

In Soar, there is no *activation* of declarative chunks and productions. In ACT-R, there is a continuously varying activation of declarative chunks and productions. The activation calculation apparatus is subsymbolic (neural network like), and uses 35 variables and parameters.[4] The activation of declarative chunks is based on its past usefulness, relevance to the general context, relevance to the specific match required, and some added noise to avoid getting stuck in local minima (Lebiere 2002). The past usefulness of a declarative chunk diminishes with time, and increases each time it is used. The activation of productions is (e) expected gain or expected utility. It depends on (q) the probability of the production working successfully, (r) the probability of achieving the goal if the production works successfully, (g) the value of the goal, (a) the cost in time that the production will take, and (b) the amount of time it will take to reach the goal after the production has been completed.[5]

$$e = q * r * g - (a + b) \qquad (1)$$

In Soar, subgoals are created automatically when an impasse occurs. Such an impasse occurs when a decision about what operator to apply cannot be made. In ACT-R, the creation of subgoals has to be specified explicitly in the action side of a production.

Soar has only one mechanism for *learning*, called chunking. Chunking is a symbolic mechanism for learning production rules. The chunking mechanism makes a new production based on the successful resolution of an impasse. "Whenever problem solving has provided some result, a new production will be created, whose actions are these just-obtained results and whose conditions are the working-memory elements that existed before the problem solving started that were used to produce the results."[6] ACT-R uses a similar mechanism for symbolic learning of productions called production compilation. This mechanism, however, is still a topic of discussion, and thus far not very well specified in the available scientific literature. The problem is that the symbolic learning of productions using Soar's chunking and of earlier versions of production compilation tended to produce too many productions. The efforts of the ACT-R group directed at limiting this proliferation of new productions are still going on. ACT-R

[4] Anderson and Lebiere, 1998, p. 434.
[5] Anderson and Lebiere, 1998, p. 61.
[6] Newell, 1990, p. 185.

has two other mechanisms for symbolic learning, both resulting in the creation of new declarative chunks. Firstly, declarative chunks can be encoded based on perception data. These encoded perception data become a repository of concrete knowledge. Secondly, the popping of a goal results in the storage of the completed goal as a declarative chunk. These popped goals become a repository of abstract knowledge.[7] In addition to symbolic learning, ACT-R has subsymbolic learning mechanisms. Without use, the activation of declarative chunks and productions diminishes gradually; they more or less fade away until they reach a level at which they no longer can be retrieved. This "forgetting" of memory elements is impossible in Soar. Based on their successful use, the activation of declarative chunks and production rules is enhanced (partially determined by a Boltzmann factor).

In recent versions of ACT-R (4.0 and 5.0), several modifications have been made. These modifications fit in a drive to make ACT-R more fine-grained, matching experimental data. One of the modifications is to limit the unrealistically powerful computations of the ACT-R architecture (Anderson and Lebiere 1998; Lebiere 2002), for instance following from the complexity of production rules that led to complex interdependent retrieval with backtracking. Other restrictions have been made to the action side of productions. Furthermore, modifications have been made to the mechanism of symbolic learning of productions. In ACT-R 5.0, the architecture has changed by distinguishing the long-term declarative and production memory modules from several short-term memory buffers. Examples of these buffers are the goal buffer, the visual buffer, and the motor buffer (Lebiere 2002).

Because of the distinction between declarative and procedural memory, the availability of perception and action buffers and of subsymbolic learning ACT-R seems to be more suitable as a basis for modelling actors than Soar.

However, ACT-R has several shortcomings as far as interaction with the environment and the handling of social behaviour are concerned.

6.2.2 Problems of traditional cognitive architectures with interaction with the environment

There is a long history of criticism on traditional cognitive architectures like Soar and ACT-R because they do not place the interaction of the actor with the environment (including other actors) centrally. There are two lines

[7] Anderson and Lebiere, 1998, p. 102 [8].

of criticism: the lack of physical grounding, and the lack of symbol grounding (Vogt 2002).

According to Simon, in his famous "ant in the sand" parable,[8] "Human beings, viewed as behaving systems, are quite simple. The apparent complexity of our behaviour over time is largely a reflection of the complexity of the environment in which we find ourselves." According to Brooks (1989), it is better to construct such an "ant in the sand" deriving its intelligence from the interaction with the environment first, in order to understand human cognition. Brooks criticizes the lack of physical grounding of traditional artificial intelligence (Brooks 1990; Brooks 1991). He wants to base a new approach in artificial intelligence on the physical grounding hypothesis. "This hypothesis states that to build a system that is intelligent it is necessary to have its representations grounded in the physical world. Our experience with this approach is that once this commitment is made, the need for traditional symbolic representations soon fades entirely. The key observation is that the world is its own best model. It is always exactly up to date. It always contains every detail there is to be known. The trick is to sense it appropriately and often enough." (Brooks 1990) Brooks furthermore states (Brooks 1991): "We hypothesize that much of even human level activity is similarly a reflection of the world through very simple mechanisms without detailed representation." Using Von Uexküll's (Uexküll and von Kriszat 1970) semiotic Umwelt concept, Brooks argues that cognition does not necessarily need internal representations of the outside world, but can use the semiotic Umwelt instead (Brooks 1990). Based on this point of view, Brooks and his collaborators have developed an alternative approach to embodied cognitive agents focusing on "developmental organization, social interaction, embodiment and physical coupling, and multimodal integration" (Brooks *et al.* 1999). Important points of view are, furthermore, that humans have no full monolithic internal models, and that humans have no monolithic control. Based on this approach, the famous Cog robot has been constructed that displays complex behaviour. The embodied cognitive agents approach also finds inspiration in the work of Gibson (Gibson 1979), and therefore has common roots with the Stamper school of organizational semiotics (Liu 2000; Stamper 2001; Gazendam 2004, 2005; Gazendam *et al.* 2003, 2005; Devlin 1991).

A second line of criticism on cognitive architectures like Soar and ACT-R focuses on the lack of symbol grounding. "Symbol manipulation should be about something and the symbols should acquire their meaning from reality" (Vogt 2002). Searle argues in his famous "Chinese Room Argument" (Searle 1980) that it is difficult to see how you can understand something

[8] Simon 1998, p. 53.

about the world based on symbol manipulation alone. Symbols that only refer to other symbols have no connection to the reality outside the world of symbols. Harnad (1990) defines the symbol grounding problem as follows: "How can you ever get off the symbol/symbol merry-go-round? How is symbol meaning to be grounded in something other than just more meaningless symbols?" Vogt (2002) recommends the use of the Peircean triad as a means to analyse and overcome the symbol grounding problem. This means that each sign should have its object and its interpretant. The link between object, sign, and interpretant should be learned by the actor.

We can now understand why the ACT-R actor is not very strong in reacting on events that happen in its environment. There are two simple learning mechanisms with respect to interaction with the environment and other actors: encoding the environment in declarative chunks and changing the activation of the encoded declarative chunks based on their use. More complex learning mechanisms like the learning of affordances (linking the recognition of an object with the possible behaviour patterns stored in the interpretant) and Peircean triads (linking object, sign, and interpretant) are still missing.

An important role in a multi-actor simulation is played by the real or simulated physical environment, especially if we want to base the actor's intelligence more on its interaction with the environment. However, building a simulated active environment that enables the simulated actor to learn is a relatively unexplored terrain. Examples are Epstein and Axtell's Sugarscape world[9] (Epstein and Axtell 1996) and the NEW TIES project (NEW TIES 2004).

6.2.3 Problems of ACT-R with social behaviour

A problem of the use of ACT-R is its focus on behaviour in Newell's the cognitive band and lower rational band (10^{-1} to 10^{2} sec).[10] The basic time unit of operation of ACT-R is 50 ms (which is the default time for a production action[11]), while the timescale of social behaviour lies in Newell's upper rational band and social band at 10^{3}–10^{7} sec (17 min – 12 days). The overall move of ACT-R in its versions 4.0 and 5.0 is towards modelling cognition at a finer grain size, which makes the problem to bridge the distance between 50 ms and the timescale of social behaviour even harder.

---

[9] See also http://www.brook.edu/es/dynamics/sugarscape/default.htm
[10] Newell 1990, p. 120.
[11] Anderson and Lebiere 1998, p. 431 [8].

The atomic components of ACT-R are relatively simple declarative chunks, goals, and productions, and there is not yet much knowledge about how to handle larger (molecular) structures like social constructs and social norms. Furthermore, the ACT-R actor is not very strong in handling multiple goals. Reasoning about survival, costs, bodily harm, and so on is missing.

Experiment*s* with ACT-R generally are based on hand-coded specifications of behaviour; the runs of this code are compared with behaviour of persons that are the subject of experimental tests. Experiments are not based on the development of emergent behaviour through interaction, and that is what we are interested in.

6.3 Potential Solutions Offered by Organizational Semiotics and Simulation of Emotion

Looking for possible solutions for these shortcomings we turn to two areas: organizational semiotics and simulation of emotion.

6.3.1 Solutions offered by organizational semiotics

Organizational semiotics (Liu 2000; Stamper 2001; Gazendam 2004; Gazendam *et al.* 2003, 2005; Gazendam and Liu 2005) offers interesting concepts in the field of the interaction of actors with their physical and social environment in the form of affordances, social constructs, and norms. One could say that the dependence of a human being on his or her physical and social environment is emphasized and analysed. The semiotic Umwelt and the information field offer concepts for segmentation of the environment in species-dependent and community-dependent sections. The language action perspective offers the concept of language-action based interaction protocols with roles for the participating actors. Dynamic semiotics offers tools for analysing the structures of actions, related actors and objects, and their encoding in messages.

If we look at the basic possibilities of the ACT-R based simulated actor to learn from its environment (encoding and change in activation of encoded chunks based on their use), it seems that we have to look for ways to encode the environment in terms of concepts like affordances, norms, social communities, interaction scripts, and messages. The learning of affordances seems to be a basic step that can be followed by the learning of the more complex structures mentioned.

As explained earlier (Gazendam 2003), the affordance mechanism is a basic mechanism for coupling the recognition of an object or situation with a possible or advisable behaviour pattern. An *affordance* is a set of properties of the environment that makes possible or inhibits activity (Gibson 1979). According to Gibson,[12] "The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill…. The medium, substances, surfaces, objects, places, and other animals have affordances for a given animal. They offer benefit or injury, life or death. This is why they need to be perceived".

Objects are discriminated because of what they afford, not as belonging to a fixed class of objects defined by its common features. The same object may correspond to several affordances. For instance, a stone may be a missile, a paperweight, a hammer, or a pendulum bob.[13] According to Stamper (2001), animals and humans develop repertoires of behaviour tuned to affordances in order to survive. If information is available in ambient light for perceiving them, affordances will be perceived. This means that affordances, being Gestalts, will be perceived rather than raw sense data…. This means that perception "… has to be a process of construction.[14]"

From a semiotic point of view, one could say that the perception of affordances is a process of construction of signs in the animal (or human) mind. This fits well in the view that all artefacts and sign structures have to be *constructed* within boundaries of reasonable computational costs (Simon 1976). These signs can be seen as *semi-indexical sign structures* (Gazendam 2003). A semi-indexical sign structure results from a process where raw data resulting from the interactions between actor and environment are unconsciously filtered, transformed, and matched to patterns (Marr 1982; Jorna 1990).

This process leads to a Peircean triad consisting of sign, object, and interpretant. The semi-indexical sign structure (*sign*) representing an affordance is caused by an object or pattern in the environment, for instance a situation (*object*). The semi-indexical sign structure is connected to its *interpretant* in the human or animal mind. An interpretant can be a feeling, or an effort to act, or a goal-oriented repertoire of behaviour that Peirce would call a *habit of action*.[15] A habit of action is a commitment to act and a connected action programme that governs the actual acting. Habits of action are mostly unconscious,[16] and can be seen as semi-indexical sign

---

[12] Gibson 1979, p. 127, 143.
[13] Gibson 1979, p. 134.
[14] Gibson 1979, p. 140, 304.
[15] Peirce 1907/1998, p. 430.
[16] Peirce 1905/1958c, p. 189 [33].

structure as well. A unit consisting of a semi-indexical representation of an affordance and its associated habit of action can be seen as a *unit of tacit knowledge*. Because affordances are specific for an animal species, tacit knowledge will be specific for an animal species as well.

6.3.2 Solutions offered by simulation of emotion

For a flexible handling of events that happen in its environment, the *simulation of emotion* seems crucial. Breazeal (2004; Breazeal *et al.* 2005) has designed a robot that is motivated by emotional drives. These drives correspond to a need to interact with people (the social drive), to be stimulated (the stimulation drive), and to occasionally rest (the fatigue drive). The emotion system of the robot is inspired by theories of basic emotions of humans. The emotional system adds effective information to incoming perceptual, behavioural, and motivational information, maintains an effective state, and subsequently determines an emotive response.

If the actor has a basic mechanism that generates emotional signals based on what happens in the actor's body and in the actor's environment, these signals can be used for an evaluation of the current goal structure and the associated actions. A switching of goals can be the result, enabling the actor to react to events in the environment and on its body. This also opens a possibility for an evaluation of possible actions based on social norms, when one of the drives of the actor is a social drive.

6.3.3 How to improve the ACT-R actor

We aim at an improvement of the interaction of the ACT-R actor with the environment and with other actors. To do this, we will begin with encoding of signals from the environment (including other actors) and the actor's own body as affordances. More complex sign structures like norms and social constructs can follow from the elaboration of the basic affordances. We will also develop an emotion subsystem that encodes body signals, adds effective tags to incoming information, maintains an emotional state, and generates a trigger to switch goals whenever the emotional state becomes unacceptable. We will also develop an active environment in which objects broadcast their characteristics to actors, and react to actor actions.

With respect to other aspects of the simulation, we have to limit ourselves to simple solutions. We will abstract from the problem of processing feature patterns perceived by the actor into Gestalts, and will assume that objects broadcast their characteristics in a way that enables the actor to classify them and develop its classification system. Furthermore, we have to restrict the unlimited possibilities of language. Language processing is not the primary subject of our simulation. At the moment we use (a) the FIPA-ACL[17] language actions, and (b) XML formats for declarative chunks (encoding information about the task environment), productions, time synchronization, messages about position and perception. We want to enrich this with more XML formats enabling messages about situations, norms, and social constructs.

6.4 Encoding the Environment and Body Signals

6.4.1 The environment

The environment of the actor has to be an active environment in order to let the actor learn from its interaction with the environment. The objects and actors broadcast their characteristics, a broadcast that can be received by the actor when it is in perception range. The objects and actors in the environment also have to react on the actions of the actor in order to generate a feedback effect. One of the tasks of the environment, including the objects in the environment, is to enforce the (simulated) laws of nature. One of the possibilities for doing that is to make the objects based on a finite state machine. Furthermore, the environment has to take care of time synchronization. Time synchronization has a technical aspect that has to do with the distribution of agents and environments over many computers. Not all computers have the same speed. This technical aspect of synchronization is best handled by a central time synchronizer on the central server of the simulation. The aspect of synchronization that has to do with the different timescales of thinking by the actors, the physical activities of actors, and the social activities of actors is best handled by a secondary time synchronizer attached to the environment.

When it comes to encoding the environment, we can look at (a) the perceived affordances (section 4.2), (b) the actions that are possible by the actor in relation to these affordances (section 6.4.3), (c) the perceived actions of other actors (section 6.4.4), (d) the perceived documents and messages

[17] FIPA-ACL is the successor of KQML. It is a standard for describing language actions in MAS. The companion of FIPA-ACL is OWL, a language and ontology describing the content of messages. OWL is the successor of KIF, DAML, and OIL. OWL uses XML. The OWL standards are very complex because they stem from many sources; the OWL documentation is hundreds of pages. Because of this complexity we do not use OWL.

(section 6.4.5), emotions (section 6.4.6), and the chunk types needed for encoding all this in ACT-R (section 6.4.7).

6.4.2 Affordances

Recall (see section 6.3.1) that the encoding of an affordance has to lead to a Peircean triad consisting of sign, object, and interpretant. The terrains, objects, plants, animals, and actors and other Peircean *objects* in the environment send signals about their characteristics to the actor. The actor can perceive these signals if it is in perception range of the sender. The signals sent by the object are encoded as a *sign* in the actor's mind, namely an ACT-R declarative chunk. To do this effectively, the actor has to invent a *name* for the object that acts as access point to the sign to be created. The sign in the actor's mind is connected to the *interpretant*, which is a collection of possible actions towards the *object*. The result of this encoding process is an *affordance sign* consisting of (a) the name of the object, (b) the class of the object, (c) individual attributes of the object, (d) place/time attributes of the object, and (e) possible actions towards the object.

There is a problem, however, when an affordance does not correspond to an object allowing one or several actions by the actor. In an ideal situation, an object (a) only allows one type of action by the actor (b), and they can be encoded together as a simple affordance. Most objects (a), however, afford many types of actions (b) (for instance, food can be picked up, eaten, and given to another actor). Some types of action by the actor (b) are related to several objects (a) (for instance, moving in a terrain with your own body towards another actor). This complicates the encoding of affordance signs as packages of recognized objects and their associated action possibilities. The most straightforward solution to this problem is to distinguish between affordances bound to the recognition of an object (with one or many actions enabled) and situation-affordances that are bound to the recognition of a situation (with one action or action script enabled; we want to avoid situations with many objects and many actions as elements that are encoded as much as possible). A similar solution has been reached in logic, where situations have been recognized as first-class objects (Barwise and Perry 1983; Devlin 1991). The recognition of situation-affordances also opens up the possibility to use case theory for analysis of (amongst others) the roles played by actors and objects in a situation.

6.4.3 Actions enabled by affordances

The actor needs at least a basic knowledge of physical affordances. Basic physical affordance classes in a simple simulated world are, for instance, terrain, object, food, plant, animal, actor, and document (NEW TIES 2004). The actor has a limited set of possible actions associated with each affordance class. An example table for a simple environment is given below (Table 1).

The actions to be taken by the actor are dependent on the social community and social situation recognized by the actor; this is elaborated below (section 6.4.4).

Table 1. Affordance classes

Action category	Actions	Affordance classes
A. Move body	Move	Body + terrain + [(towards/ from) something (actor, thing, food, structure, animal, document)]
	Rest	Body
B. Manipulate object	Take, Put Make	Thing|food Thing|food
	Eat	Food
C. Manipulate terrain	Make road Make house	Terrain + structure
	Make food	terrain + food
D. Coordinated behaviour with animals	Use animal	Animal
E. Coordinated behaviour with other actors (social action)	Give, receive	(Other) actor + thing|food|document
	Move, dance, work together Speak, listen, make nonverbal sign, perceive nonverbal sign Write, read	(Other) actor + terrain + body (Other) actor + message Document

Based on experimentation and experience, the actor should be able to create more complex action patterns. In a similar way, the actor should be able to develop affordance subclasses. This requires the stimulation of playful behaviour including evaluation moments by the emotion subsystem, and a correct functioning of the production compilation in ACT-R.

6.4.4 The actions of other actors

Organizational semiotics requires the recognition of the current social context in order to be able to determine the advisable behaviour in a certain situation. The current social context consists of a combination of the current social community, the type of social situation, own role or task in the current situation, and the phase of that situation related to a ritual, protocol, or script.

It is assumed that the actor determines the current social context based on the characteristics and actions of the other actors present, and – if no other actors are present – based on its awareness of its own social context. The social community can be given a name and is then encoded as an affordance sign of the type "social construct". The same holds for the type of situation, the actor roles, the script, and the phase of the script. What is a script? A script consists of phases that are linked to each other. For each phase, actor roles are distinguished. For each role in a certain phase, norms apply. A script generally is only applicable in a specific type of social situation that occurs in a specific social community.

Social constructs and norms are not encoded directly based on perception. They result out of an elaboration of newly perceived affordances related to other actors. An actor may observe that there is a usual way of doing expressible as a script in a certain situation. For instance, actor with role X does action a, then actor with role Y does b, and so on. For different situations and different communities, different scripts are applicable. Based on an evaluation of these observations, the actor may assign a *name* to a situation type, *assign* this situation type the status of a social construct, and *encode* the usual way of doing it attached as a script. In other words, the elaboration of observations related to other actors consists of reviewing these observations, identifying and naming situation types, and attaching statuses and associated scripts to these situation types. Scripts can be split in phases. Norms can be encoded for each combination of phase and actor role in which the actions to be taken or avoided are specified. Norms are encoded based on the format described by Liu *et al.* (2001; Liu 2000), in a way that in the action part of the norm there is only one deontic operator and only one action. If an actor deduces that some other actors are part of a certain social community, and this community has been given a name (possibly by himself) the community receives the status of a social affordance.

6.4.5 Documents and messages

Documents are sign structures that have a more or less autonomous existence, for instance a book, a letter, an email message. Documents are also physical affordances because they allow actions like picking up, giving, writing, and reading. Documents have to be decoded (read and understood) by the actor before they can be encoded as ACT-R chunks.

At the moment the actors (as well as the environment) use FIPA-ACL language actions, and some simple XML message formats for encoding content. In order to enable messages about situations, norms, and social constructs we will enrich the XML message format library with formats based on a set of cases or roles taken from case theory,[18] and on a limited English vocabulary reflecting the simple world we try to model (something like described in (NEW TIES 2004)). For each task environment, a suitable vocabulary will have to be created.

6.4.6 Emotions

The emotion subsystem encodes body signals, maintains an emotional state, monitors relevant incoming information, encodes emotional signals as emotion affordance signs, which enables the triggering of goal switching whenever the emotional state becomes unacceptable. For the purpose of the simulation, we follow Breazeal (2004; Breazeal *et al.* 2005) in choosing the drives *social*, *stimulation*, and *fatigue*. We add *safety* (i.e., avoid collisions). There is a subsumption hierarchy (Brooks 1989) of these drives, in which the most basic drive has the highest priority. The most basic drive is safety (safe vs. fearful), then follow (in that order) fatigue (energetic vs. tired), social (content vs. unhappy), and stimulation (excited vs. bored). The actor has at least a basic knowledge of desired actions in relation to these drives (for instance, like depicted in Table 2); based on experience this knowledge may develop in a more specific direction.

The actor's body generates signals about several bodily functions. These signals are encoded as affordance signs. The emotion subsystem maintains an emotional state and changes this emotional state based on a monitoring of relevant incoming information about performed actions and possible threats. The emotion subsystem reports the emotional state by sending signals

[18] See for instance Gazendam 1993. p. 68–75; Andersen 2004.

Table 2. Drives and actions

Drive	Actions	State indicator change
Safety +	Move (avoid collision)	Towards safe
Safety –	Move (towards something)	Towards fearful
Fatigue +	Eat|rest	Towards energetic
Fatigue –	Any action	Towards tired
Social +	Social action in accordance with norm	Towards content
Social –	Social action not in accordance with norm non-social (individual) action	Towards unhappy
Stimulation +	Do task|explore any action|evaluate and learn	Towards excited
Stimulation –	Do not do task|do not explore|do not evaluate and learn	Towards bored

consisting of a set of (three) emotional affordance signs, each consisting of (a) name, (b) drive, (c) strength of the drive indicator, (d) time, and (e) desired actions. Whenever the emotional state becomes unacceptable, a trigger is generated to switch goals.

6.4.7 Chunk types

The encoding (including elaboration) leads to the following chunk types to be implemented in ACT-R declarative memory:

1. Affordance (sign)
    1.1. Physical affordance (sign)
        1.1.1. Physical situation (sign)
        1.1.2. Document (sign)
    1.2. Emotion affordance (sign)
    1.3. Social construct
        1.3.1. Social community
        1.3.2. Social situation
        1.3.3. Actor role
        1.3.4. Script
        1.3.5. Phase in script
2. Norm
3. Element of affordance
    3.1. Name
    3.2. Class or, (in case of emotion affordance) drive

    3.3. Attributes (in case of emotion affordance, emotional state indicators)
    3.4. Place and time attributes
    3.5. Possible actions or, (in case of emotion affordance) desired actions
    3.6. Norms (in case of social construct)
4. Element of norm
    4.1. Name
    4.2. Type
    4.3. Context = social community, situation, own role or task in situation, phase in script
    4.4. Condition
    4.5. Deontic operator
    4.6. Action

6.5 Using the Encoded Affordances, Emotions, and Norms

6.5.1 Awareness

The subsystem that implements the actor's awareness of its environment and its emotional state consists of an awareness buffer and an awareness handler. The *awareness buffer* contains information about the current situation in terms of the current social context and the current emotional state. The awareness buffer is necessary to select the social affordances and norms that are used for the determination of an actor's behaviour. The *awareness handler* has the task to trigger the elaboration the new affordance information in the perceptual buffer and the subsequent integration of the elaboration results in declarative memory, to update the awareness buffer, and to generate a *trigger to change goals* if that is necessary. As soon as one of the emotional state indicators sinks below the acceptable level, the awareness handler generates a trigger to determine what goal has to get focus.

The actor system has a basic loop, in which (1) the body and the environment of the actor put new information encoded as affordance signs in the perceptual buffer, (2) the emotion subsystem encodes the emotional state, (3) the awareness handler gets a chance to update the awareness and eventually trigger a goal change, and (4) the goal handler gets a turn to select and fire productions.[19]

[19] This basic loop is controlled by the internal clock of the actor.

6.5.2 Goal switching based on emotion

The overall goal of the ACT-R actor could be a combination of:

- Change the emotional state to an acceptable level.
- If the emotional state is acceptable, look for further improvement of it.
- Search for an action sequence that leads to the desired emotional state in an acceptable time.
- If the emotional state is perfect, rest a while or explore a while.

If a trigger is generated because the emotional state has become unacceptable, the goal stack is emptied until this overall goal gets focus again. The popped goals are remembered in declarative memory. They will be activated again as soon as changes in the emotional state make them urgent.

6.5.3 The use of social constructs and norms

In the declarative memory, a collection of chunks representing possible actions and their direct effects on the emotional state is maintained (this could, for instance, be a representation in chunks of Table 2). In the awareness buffer, the current context is maintained. Using the current context, the social constructs and associated norms that are applicable in the current context can be determined.

If an applicable norm discourages (as forbidden) a certain action, application of that action would lead the emotion subsystem to change the emotional state with respect to the social drive in a negative direction. Much in the same way, actions that are encouraged by the applicable norms will change the emotional state in a positive direction. A production that captures this knowledge could read something like: “If your goal is to maintain or improve your emotional state, then avoid actions that predictably lead to a deterioration of the emotional state, and perform actions that predictably lead to an improvement of the emotional state”. More specific productions regarding the effects of actions on the emotional state must be learned by the actor based on trial and error.

6.5.4 Emotional stimulation of task performance, learning and social behaviour

The stimulation drive encourages doing tasks, trying out actions (by exploration and imitation), evaluation, and subsequent learning. The social drive encourages social behaviour by an emotional premium on behaviour according to norms.

6.5.5 Learning

Learning in the ACT-R architecture results from the use of the encodings discussed above. In a multi-actor system, an ACT-R actor could learn based on:

1. *Exploration*: encoding environment signals and signals from the body system, trying out actions and action patterns
2. *Evaluation*: identifying of, and generating names for, successful action patterns and subclasses of affordances
3. *Imitation*: encoding patterns of interaction (including communication) between actors as scripts with roles and imitation of these patterns in the appropriate situations
4. *Knowledge transfer*: decoding and encoding messages and documents
5. *Social conflict resolution*: storing solutions to social conflicts as social constructs (Gazendam 2005)

6.6 Conclusion

The architecture of human cognition ACT-R can be enhanced by the encoding of the affordances, social constructs, and norms that the environment offers. Another possible improvement in ACT-R is goal switching based on emotions and awareness of the current situation. An emotion subsystem can encourage task performance, learning, and social behaviour. In this way, a bridge could be built between mainstream science emphasizing the inner workings of the cognitive system, and organizational semiotics that emphasize the dependence of an actor on the physical and social environment.

The investigation into the use of concepts from organizational semiotics makes clear that the usefulness of concepts like affordance, norm, situation, and script can be improved if they are redefined in a more precise way using knowledge about the functioning of the human cognitive system.

References

Andersen PB (2004) Anticipated activities in maritime work, process control, and business processes. In: Liu K (ed.), Virtual, Distributed and Flexible Organisations: Studies in Organisational Semiotics, Kluwer Academic, Dordrecht, The Netherlands, pp. 35–59

Anderson JR (1990) Adaptive Character of Thought, Benjamin Cummings, Redwood City, CA

Anderson JR, Lebiere C (1998) The Atomic Components of Thought, Lawrence Erlbaum, Mahwah, NJ

Barwise J, Perry J (1983) Situations and Attitudes, Cambridge, MIT Press, MA

Breazeal C (2004) Function meets style: insights from emotion theory applied to HRI. IEEE Transactions on Systems, Man and Cybernetics, Part C, 34, pp. 187–194

Breazeal C, Buchsbaum D, Gray J, Gatenby D, Blumberg B (2005) Learning from and about others: towards using imitation to bootstrap the social understanding of others by Robots. Artificial Life, 11, pp. 31–62

Brooks RA (1989) How to build complete creatures rather than isolated cognitive actors. In: VanLehn K (ed.), Architectures for Intelligence, Lawrence Erlbaum, Hillsdale, NJ, pp. 225–239

Brooks RA (1990) Elephants don't play chess. Robotics and Autonomous Systems, 6, pp. 3–15

Brooks RA (1991) Intelligence without representation. Artificial Intelligence Journal, 49, pp. 139–159

Brooks RA, Breazeal C, Marjanovic M, Scasselati B, Williamson MM (1999) The Cog Project: Building a Humanoid Robot. Lecture Notes in Artificial Intelligence 1562, Springer, New York, pp. 52–87

Devlin K (1991) Logic and Information, Clarendon University Press, Cambridge

Epstein JM, Axtell R (1996) Growing Artificial Societies: Social Science from the Bottom Up, MIT Press, Cambridge, MA

Gazendam HWM (1993) Variety Controls Variety: On the Use of Organization Theories in Information Management, Wolters-Noordhoff, Groningen, The Netherlands

Gazendam HWM (2003) Models as coherent sign structures. In: Gazendam HWM, Jorna RJ, Cijsouw RS (eds.), Dynamics and Change in Organizations: Studies in Organizational Semiotics, Kluwer Academic, Dordrecht, The Netherlands, pp. 183–213

Gazendam HWM (2004) Organizational semiotics: a state of the art report. Semiotix, 1(1). Retrieved 27 May 2005 from http://www.semioticon.com/semiotix

Gazendam HWM (2005) Coordination mechanisms in multi-actor systems. In: Jorna RJ, van Wezel W, Meystel A (eds.), Planning: Its Aspects, Motivations, and Methods, Wiley, New York

Gazendam HWM, Liu K (2005) The evolution of organisational semiotics: a brief review of the contribution of Ronald Stamper. In: Filipe J, Liu K (eds.), Studies in Organisational Semiotics, Kluwer Academic, Dordrecht, The Netherlands

Gazendam HWM, Jorna RJ, Kecheng. L (2005) Organizational Semiotics: Round Table Workshop "An Organizational Semiotic View on Interculturality and Globalization". Proceedings of the IASS 2004 Conference

Gazendam HWM, Jorna RJ, Cijsouw RS (2003) Introduction. In: Gazendam HWM, Jorna RJ, Cijsouw RS (eds.), Dynamics and Change in Organizations: Studies in Organizational Semiotics, Kluwer Academic, Dordrecht, The Netherlands, pp. 1–12

Gibson JJ (1979) The Ecological Approach to Visual Perception, Houghton Mifflin, Boston, MA

Harnad S (1990) The symbol grounding problem. Physica D, 42, pp. 335–346

Helmhout M, Gazendam HWM, Jorna RJ (2004) Social constructs and boundedly rational actors: a simulation framework. In: Liu K (ed.), Virtual, Distributed and Flexible Organisations: Studies in Organisational Semiotics, Kluwer Academic, Dordrecht, The Netherlands, pp. 153–179

Helmhout M, Gazendam HWM, Jorna RJ (2005a) The role of organizational semiotics and social constructs in the social awareness of simulated cognitive plausible actors. In: Filipe J, Liu K (eds.), Studies in Organisational Semiotics, Kluwer Academic, Dordrecht, The Netherlands

Helmhout M, Gazendam HWM, Jorna RJ (2005b) Emergence of Social Constructs and Organization: A Cognitive Learning Approach. Proceedings of the 21st EGOS Colloquium, 30 June to 2 July, Berlin

Jorna RJ (1990) Knowledge Representation and Symbols in the Mind, Stauffenburg Verlag, Tübingen, Germany

Lebiere C (2002) Introduction to ACT-R 5.0: Tutorial 24th Annual Cognitive Science Society, 8–10 August 2002. Retrieved 7 June 2005, from http://actr.psy.cmu.edu/tutorials/ACT-R_intro_tutorial.ppt

Liu K (2000) Semiotics in Information Systems Engineering, Cambridge University Press, Cambridge

Liu K, Sun L, Dix A, Narasipuram M (2001) Norm-based agency or designing collaborative information systems. Information Systems, 11, pp. 229–247

Marr D (1982) Vision, Freeman, New York

NEW TIES (2004) New and Emergent World Models Through Individual, Evolutionary, and Social Learning: Annex I Description of Work (Proposal/Contract No.: FP6-502386, 5 April 2004), European Commission, Brussels. Retrieved 9 June 2005 from http://www.cs.vu.nl/%7Egusz/newties/annex1-short.pdf

Newell A (1973) Production systems: models of control structures. In: Chase WG (ed.), Visual Information Processing, Academic Press, New York, pp. 283–310

Newell A (1990) Unified Theories of Cognition, Harvard University Press, Cambridge, MA

Peirce CS (1958) What pragmatism is. In: Wiener PP (ed.), Charles S. Peirce: Selected Writings: Values in a Universe of Chance, Dover, New York, pp. 180–202 (Original work published in The Monist, 15, April 1905, pp. 161–181)

Peirce CS (1998) Pragmatism. In: Houser N, De Tienne A, Eller JR, Clark CL, Lewis AC, Davis DB (eds.), The Essential Peirce: Selected Philosophical Writings: Vol. 2 (1893–1913), Indiana University press, Bloomington, IN, pp. 398–433 (Original work published 1907)

Roest GB (2004) Rbot-Project. Unpublished thesis, Hanzehogeschool, Groningen, 7 June

Searle JR (1980) Minds, brains and programs. Behavioral and Brain Sciences, 3, pp. 417–457

Simon HA (1976) Administrative Behavior: A Study of Decision-making Processes in Administrative Organization, 3rd edn, The Free Press, New York (Original work published 1945)

Simon HA (1998) Sciences of the Artificial, 3rd edn, MIT Press, Cambridge, MA (First edition published 1969)

Stamper RK (2001) Organisational semiotics: informatics without the computer? In: Liu K, Clarke RJ, Andersen PB, Stamper RK (eds.), Information, Organisation and Technology: Studies in Organisational Semiotics, Kluwer, Boston, MA, pp. 115–171

Uexküll J, von Kriszat G (1970) Streifzüge durch die Umwelten von Tieren und Menschen. Fischer, Frankfurt, Germany (Original work published 1936)

Vogt P (2002) The physical symbol grounding problem. Cognitive Systems Research Journal, 3, pp. 429–457

Part 4
Transformation of Information

Chapter 7
Semiotic Transformation from Business Domain to IT Domain in Information Systems Development

M. Gan[1], K. Liu[1,2], B. Han[2]

[1]*School of Management and Economics*
Beijing Institute of Technology, China
[2]*Informatics Research Centre*
The University of Reading, UK

Abstract

Semantic transaction loss exists in terms of concepts transformation from one design stage to another in information systems development. It results from different interpretations and representations of various requirements in design domains. Without an explicit structural specification of semantic linkages among design domains, the transformation cannot be efficiently identified in an appropriate way. In this chapter, a mechanism for transformation is proposed to assist with the problem through connecting different aspects of information systems with a precise and coherent representation. For the mechanism, transformation begins with the analysis of business objects in business domain, and finishes by generating corresponding structural components in IT domain. Components and their relationships in each domain are endowed with correlated semantic interpretation. The processes of transformation are illustrated through signs and their structure in an organizational semiotic perspective.

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 133–143.

7.1 Introduction

Information systems development can be seen as a series of semiotic transformations across business and IT domains. But the transformations usually proceed separately to keep integrity in each process. However, concept divergence, which refers to the incoherence of concept structure in the processes, often exists and causes many problems. One of the serious problems is called transaction loss which happens during transformations among processes of system analysis, system design, and system implementation. When a model in business domain needs to be transformed to IT domain, semantic incoherence will be introduced during the transformation. This problem will consequently cause inconsistency between the evolution of the information system and the business change.

The semantic barrier is summarized by Martin *et al.* (2002) on information systems development from an organizational semiotics point of view. Organizational semantics, software issues, and semantic gaps are the three aspects of the semantic barrier. It is said that the capability to understand the different perceptions of others is quite important for people to share knowledge in system development from the perspective of organizational semantics. The software based tools which represent and model the business also have semantic problems in terms of integration with other systems and with people. Thus, the semantic barriers reveal a gap between the developers of systems and the real world of the organization. Defining and formalizing different levels of a shared conceptual understanding and what it represents in the changing business context is a major challenge in system development.

The problem is addressed in this chapter through introducing a mechanism for transformation from business domain to IT domain. The mechanism acts as a set of signs for a desired mapping. The study is inspired from the theory and technologies of convergent engineering (Taylor 1995) which tries to combine together traditionally separate development processes by converging them through transforming organization, process, and resource (OPR) (Hubert 2001).

7.2 Literature Review

7.2.1 Organizational semiotics

Semiotics is the science which studies the phenomena of signification, meaning and communication in natural and artificial systems (Nöth 1995).

Its main artefact is the notion of signs, and its main approach is to explain different kinds of phenomena as being sign processes (Gudwin 2004). Both natural and artificial systems can be modelled semiotically.

When it comes to signs, it is actually about the meaning it conveys to different users (i.e. model designers) who encode the meaning in a model design process (Xu and Feng 2004). In order to make a conversation or communication with others, the meaning of the signs being used must be shared, although it could be only part of the semantic information carried by the sign or the pragmatic meaning of the sign.

The contribution of the semiotic perspective is on three aspects (Connolly and Phillips 2002). First, it makes it clear how a single term can mean different things in relation to different levels in the hierarchy. Second, in turn it helps to organize our thinking when designing or evaluating a system, motivating us to consider the implications of design principles at the various levels of syntactics, semantics, and pragmatics. Third, it is important that interlevel relationships need to be given consideration.

The theory of semiosis shows the sign could be anything that refers to another concept other than itself and the linking between the *sign* and the *object* indicates their relationship (see Fig. 1). The following triangle shows that the interpretant of the signs enables transformation between objects and signs (Peirce 1960 cited by Stamper *et al.* 2000; Liu *et al.* 2002a; Gudwin 2004).

Different aspects of information systems can be regarded as the sign systems, ranging from pure technical to social and organizational issues (Goldkuhl and Ågerfalk 2002). Organizational processes can be described in terms of sign processes, which is the main idea behind organizational semiotics.

7.2.2 Convergent architecture

Convergent Engineering (CE) is introduced by Taylor (1995) as a design vision with a methodology which aligns business and IT systems through a

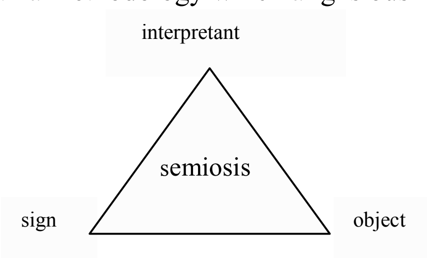

Fig. 1. A version of Pierce's semiosis triangle

set of design patterns and design techniques. Both the business and the IT perspectives of a system can be described in one model.

MDA stands for model driven architecture introduced by object management group (OMG) (Miller and Mukerji 2003). Convergent Architecture (CA), as a successful practice of CE, is an MDA-centric IT-architectural style. It leverages the benefits of agile development and CE as well as signifies the alignment of business and IT models into one common, synchronized model. CA provides a practical, architecture-driven MDA process and techniques enabling business and software design to become a consolidated effort.

Key technologies and theories of CA are as follows (Hubert 2001):

Technology Projection (TP) is a feature of an MDA-centric approach as well as an IT-architectural style which transforms a model into another or as a final step, maps a model to a workable system infrastructure.

Reduced abstraction set computing (RASC) represents the core component abstractions and corresponding architectural layers to achieve the goals of CE, through which the design processes of the holistic system are consolidated together.

OPR are the three core abstractions of RASC and are verified to be the minimum set of the formal structural components through which business model can be transformed to IT system. OPR are also the design pattern introduced by CE.

7.3 A Semiotic View of Information Systems Design

An assumption is that both business domain and IT domain can be separately abstracted as a certain set of structured signs being OPR, which enables the transformation. The intention of the purposive activities in the mechanism is to make the processes coherent semantically and consolidate them as a whole to produce integrity in domains. (see Fig. 2)

7.3.1 Transformation processes of sets of signs

Business domain can be regarded as a set of *objects* (in semiosis I) with a certain configuration. In information systems design, these *objects* will be abstracted into a business model as a set of *signs* (in semiosis I) with certain concepts. This process of transformation can be viewed as from *objects* (business objects) in business domain as real world to a corresponding set of *signs* (as images of the objects) in business model. It models the business

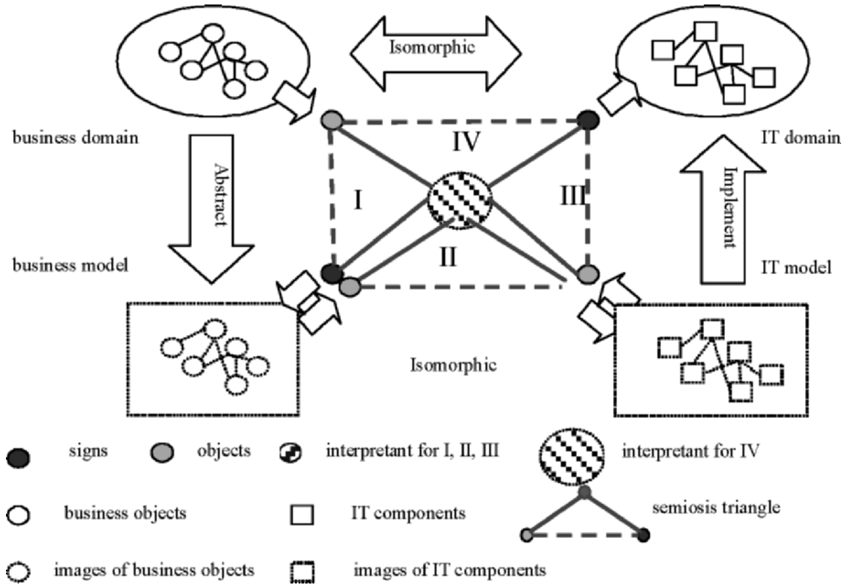

Fig. 2. The semiosis view of isomorphic transformation between business and IT domains

world through *interpretant* (in semiosis I) and represents the process of system analysis. Isomorphism means coherence between business domain and the description (i.e. the model) of it in business model.

When business model is achieved as a set of *signs* (in semiosis I) through transformation, it can be also viewed as a set of *objects* (in semiosis II) in the process from business model to IT model. Then IT model is achieved as *signs* (in semiosis II) through the mechanism of isomorphic transformation which guarantees the isomorphic configuration of both sets of signs (objects) in business model and IT model. A sign in each model can be transformed into another corresponding sign in the other context. *Interpretant* (in semiosis II) is represented and explained through the mechanism. This process of transformation represents system design and can be viewed as from *objects* (in business model) to *signs* (in IT model) through *interpretant* (the mechanism of isomorphic transformation).

In IT model, the configuration of the set of signs reflects components and their structure in IT domain. In the process from IT model to IT domain, the set of signs in IT model achieved in the process I acts as a set of *objects* (in semiosis III) being transformed through *interpretant* (in semiosis III) to a set of *signs* (components and their structure) (in semiosis III) in IT domain. This process represents system implementation. IT domain is regarded as *signs* and the images of IT domain (IT model) are regarded as *objects* in semiosis II.

7.3.2 Isomorphic structure for convergent architecture

Finally the set of *signs* (in semiosis III) in IT domain reflect the realization of the set of *objects* (in semiosis I) in business domain. When the two domains are focused on through hiding the other two models being images of them, it can be found in semiosis IV that the set of *objects* (in semiosis IV) is transformed to the set of *signs* (in semiosis IV) with an isomorphic structure from business domain to IT domain through *interpretant* (in semiosis IV) (the combination of the three former processes of transformation). Thus the convergent architecture can be obtained with the isomorphic structure where one model (set of signs) reflects both business and IT domain. The design pattern is described as the union set of signs.

Thus the whole transformation process finishes with sets of isomorphic structured objects (components) in both domains. Since the three sets of signs in different contexts have same structure and each sign can find a corresponding sign in every other context, a union set of signs and its configuration can be used to represent these different sets of signs. The combination of the three interpretants is described as the one in semiosis IV. In the combined interpretant, the union set of signs acts as rules for the transformations between contexts. It is called the isomorphic structure of domains.

The semiosis processes enable the mechanism of concept transformation in domains as well as concentrate analysis of semantic in each context. Thus, whenever changes happen in business domain by adding, deleting, or adjusting business objects or their structures, corresponding changes in IT domain can follow them to realize a synchronizing coherence between both the domains. The mechanism of this implementation is maintaining isomorphic structure in sets of concepts in domains and models through transforming *objects* to *signs* with a continuous feedback loop. When we get the information system in IT domain, the transformation processes can be utilized in the other direction (from *signs* to *objects* in all the four semiosis triangles) through which the verification of the coherence between domains can be realized. The other parts of information system development such as assembly testing, integration testing, and user acceptance testing will be accomplished by following this direction of transformation processes and the theories of loose-coupling and reduced abstract set computing. Thus the transaction loss between different transformations can be solved consequently based on the isomorphic structure.

To realize the transformation, in sections 4 and 5, it is focused on the realization of isomorphic structure consisting of OPR and the transformation mechanism based on them on configuration and is analysed and explained from organizational semiotics point of view.

7.4 The Signs and Structure of OPR

In business domain and IT domain, objects and components can be abstracted as a set of signs which consists of three elements: OPR. Each element has its own hierarchical structure separately, which can be viewed as a sign representing an object (component). Each of the three represents a corresponding aspect of business objects or IT components using signs.

7.4.1 Loose-coupling

The structure of the signs refers to the pattern with which OPR are interacted. It is not only used to explain, abstract and refine core business objects in a logical way, but also used to configure IT infrastructure through the way business objects are mapped to technological components with loose-coupling. Loose coupling is a dialectical concept in organization theory that emphasizes the simultaneous existence of rationality and indeterminacy in a system. Loose coupling between systems implies the existence of elements that are linked ("coupled") to preserve some degree of determinacy. At the same time, the elements are subject to spontaneous change, leading to some degree of independence ("looseness"). Loose coupling reduces interdependencies, allowing organizational components to more easily deal with change, and it also makes it easier for them to be disentangled and recombined into new configurations. This is expected to result in both offering flexibility and partnering flexibility.

7.4.2 Interaction of OPR

The interaction among OPR elements is focused on every single layer of their hierarchical structure. At every single layer, each element has basic affordances with each other (see Fig. 3): organization manages process and resource. Process utilizes and generates resource while resource manages itself. Based on object-oriented technology and theory, objects (components) can be illustrated through inherence, encapsulation, and polymorphism of OPR at one layer, which can be given certain characters from upper layers where their subclasses exist as OPR as well. This structure can be described in IT domain with the isomorphic structure and can be mapped to different realizations according to a certain technology projection.

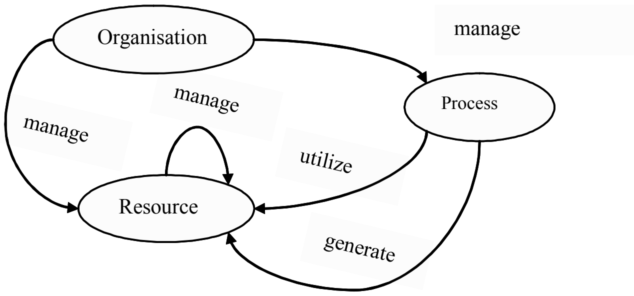

Fig. 3. The relationship between OPR

Since OPR in business domains and IT domain can be described in the same way. Their structure can help different sets of signs in domains following a same structure to organize the relationship of components. Thus it connects the two domains together with a union structure of signs and makes components in both domains interact with each other. The structure of OPR also affects significantly the performance of flexibility and agility in information systems. The rules for interaction among different sets of signs are defined, which will make behaviour of the components in both domains follow the same structure of OPR without supervising every component individually.

7.5 Realization of the Transformation Mechanism

OPR can keep the whole development process follow the same isomorphic structure. Norm analysis (Liu 2000) helps keep the isomorphic structure in the design processes and maintain the semantical integrity to reduce transaction loss. The construct of a norm has several elements:

- <condition>: defines context and specify triggers.
- <agent>: an individual member or a group in an organization.
- <action>: links to process and can be categorized as several types of subactions.
- <D: a deontic operator to describe the responsibility such as obligation, permission, and prohibition.

7.5.1 Norm analysis for OPR structure

The semiotic structure of a norm, <condition>→<D><agent><action>, can be explained by the design pattern of OPR. <action> is linked to process, while <agent> is part of the organization, and <action> can be categorized as several types of subactions. *Knowing* and *doing* are the top categories of the action. Within *doing* action, *utilizing* and *producing* are the second level categories of the action. Resource is always associated with <action> as signs and objects. The deontic operator with <agent> in every <action> is labelled by <D> with certain responsibilities of OPR. The context and trigger is labelled by <condition>.

The responsibility of a certain object class comes from the analysis of business objects consisting OPR. <Agent> inherits all the responsibilities of its subclasses and is endued with the responsibilities of its own object class. The responsibilities are explained using Norm Analysis.

7.5.2 TPC and interfaces

Structure configuration of IT components and their communication patterns can be transformed as a set of structured signs. It shows that business domain and IT domain have close relationship with each other. Later, the detailed description of the interfaces from business domain to IT domain with the set of signs will be obtained through using TPC reflecting the mechanism of transformation. Each interface represents a certain interchangeable process in different contexts by applying meta-model of the signs to map different models.

TPC is used to map business objects to IT domain through signs, which is the abstraction of the technology project. It shows different presentations according to different technology project in order to adapt to the probable platform in future. TPC constitutes the core of UML (OMG/MDA). The challenge TPC meets is supporting robust and automatic mapping to available technologies based on standard technologies such as J2EE/EJB and reducing limitation of OPR in business domain at the same time.

7.6 Conclusion and Future Work

This chapter analyses and explains the transformation processes from business domain to IT domain in information systems development from a semiotic perspective. A mechanism is proposed to enable images in business model and IT model to be isomorphic. It makes objects (components) in business domain and IT domain follow an isomorphic structure to enable synchronizing changes in the two domains. With an isomorphic concept structure of different sets of signs, the mechanism for interaction and corresponding between the two domains will be obtained. This mechanism is also supposed to be an effective approach to reduce semantic transaction loss.

OPR is introduced as core elements in every domain as well as the union set of signs in every model. Thus these models can be transformed into each other to reduce the transaction loss among different processes, which can be clearly connected and described by isomorphic structured objects (components). The automatic transformation between business and IT domains in information systems development can be put into reality through using technology project component.

This solution leads the design of information systems to be a synchronized process and produce an isomorphic concept structure to avoid concept divergence in domains (models). Systems which are designed using the isomorphic structure are proposed to be understandable, maintainable, and easily modified in response to the changing business conditions.

Following the current development, future work will focus on the implementation of the transformation mechanism and the hierarchical structure of OPR in both domains as well as the isomorphic structure of their configuration. To put it into reality, technology project component will be discussed to support the realization, with which the automatic transformation is proposed through using the mechanism introduced. Transaction loss will be analysed and solved at two aspects which are design method and design process of an information system. The mechanism is achieved through using several transformation rules among sets of signs in information systems design, which define constraints, conditions, and policies for how the components are configured and assembled. A schema is required to model the relationships among OPR elements for transformation based on the mechanism, which requires establishing rules for specifying relationships and assembling a set of objects (components) to achieve certain goals in business (IT) domain. On the other hand, actors and their responsibilities in information systems development will be discussed with Norm Analysis to guarantee the realization of the transformation mechanism.

Acknowledgement

This work is completed during Mingxin Gan's academic visit to the Informatics Research Centre at The University of Reading. The authors would like to express sincere thanks to Dr. Lily Sun and Prof. Steef Peters for their encouragement, support, and valuable suggestions.

References

Connolly JH, Phillips IW (2002) User-system interface design – an organisational semiotic perspective. In: Liu, K *et al.* (eds.), pp. 119–132

Goldkuhl G, Ågerflk P (2002) Actability: A way to understand information systems pragmatics. In: Liu, K *et al.* (eds.), pp. 85–114

Gudwin RR (2004) Semionics a proposal for the semiotic modeling of organisations. In: Liu K (ed.), pp. 15–34

Hubert R (2001) Convergent Architecture: Building Model-Driven J2EE Systems with UML. Wiley, New York Liu K (2000) Semiotics in Information Systems Engineering. Cambridge University Press, Cambridge

Liu K, Liao SY, Chong S (2002a) Semiotics for Information Systems Engineering-Re-Use of High-Level Artifacts, Proceedings of 3rd International Symposium on Communication Systems, Networks and Digital Signal Processing, Stafford, Virginia, 1–4

Martin VA, Lycett M, Macredie R (2002) Integrating Information Systems-A Semiotics Approach. In: Liu, K, Clarke, RJ, Andersen PB, Stamper RK (eds.) pp. 109–118

Miller J, Mukerji J (eds.) (2003) MDA Guide Version 1.0.1, OMG Document: omg/2003-06-01, http://www.omg.org/docs/omg/03-06-01.pdf

Nöth W (1995) Handbook of Semiotics. Indiana University Press, Bloomington, IN

Peirce CS (1960) Colleted Papers of CS Peirce, Hartshorne C, Weiss P (eds.), Harvard University Press, Cambridge, MA

Stamper R, Liu K, Hafkamp M, Ades Y (2000) Understanding the Roles of Signs and Norms in Organizations–A Semiotic Approach to Information Systems Design. Journal of Behaviour and Information Technology 19(1), pp. 15–27

Taylor DA (1995) Business Engineering with Object Technology. Wiley, New York

Xu H, Feng J (2004) A semiotic insight into model mapping. In: Liu K (ed.), 293–302

Chapter 8
Comparative Analysis of Ontology Charts and other Modelling Techniques

J. Cordeiro, J. Filipe

Escola Superior de Tecnologia de Setúbal, Rua do Vale de Chaves, Estefanilha, 2910-761 Setúbal, Portugal

Abstract

Ontology charting (OC) is an organisational semiotics (OS) technique for the representation of the requirements of organisational information systems, based on semiotic theories, incorporating technical and social aspects. The application of semiotics to the design of information systems (IS) competes with other methods. This chapter presents a comparative analysis of the modelling techniques used by some of these methods, applied to a case study, discussing each model's characteristics and expressive power. A framework to guide this comparison is also introduced.

8.1 Introduction

Modelling plays a major role in information systems development (ISD). As any language, modelling languages determine the way we perceive, plan, and act in the ISD world. In ISD, as in many other engineering areas, models are often expressed by diagrammatic languages or diagrams presenting some basic elements and their relationships. These elements are usually associated with the key concepts of the underlying theories and methodologies. Moreover models can be considered *simplifications of the reality,* therefore it results that by analysing model elements we are also analysing the related theories, their relevant concepts and their interpretation of the reality. In this sense, by studying and comparing models it is possible to understand the

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 145–163.

philosophical foundations and the particular perspectives used by each methodology and adopted by each theory. In this chapter different kind of diagrams and models will be analysed and compared revealing approaches, focus, and missing concepts of their supporting theories.

A first kind of diagram and the centre of our analysis are ontology charts (OC) used by Stamper's theory of organisational semiotics (Stamper 1973; 1996; 2000). OCs permit to model organisations and can be used for driving the analysis and design of information systems by offering a stable and precise view of organisational requirements. Stamper's OS theory provides as well a set of methods for requirements analysis: the methods for eliciting, analysing, and specifying users' requirements methods (MEASUR). These methods include: problem articulation methods (PAM), semantic analysis method (SAM), and norm analysis method (NAM) – that enable us to capture issues from a vague, complex problem, to assist the problem-owner in eliciting and to specify the general patterns of behaviour of the agents in the business system (Liu 2000). Particularly SAM, departing from the terms used in the problem statement, establishes a sequential set of steps that will have as a deliverable one or more OCs containing the requirements model or ontological schema.

The other kinds of diagrams analysed are dynamic essential modelling of organizations models as a particular application of the Language Action Perspective theory (see Winograd and Flores 1986 and Reijswoud and Dietz 1999); role-activity diagrams applied by the Riva method (Ould 2005), DIPLAN used in the Theory of Organized Activity (Holt 1997), and UML activity diagrams (Booch *et al*. 1998) for business process modelling. These diagrams or modelling techniques and associated methods or theories share as a common basis their suitability to model business and organization processes, as well as their human, social, and organizational nature, without concerning necessarily the supporting information technology (IT).

All the target diagrams and models and the related theories are summarized and presented in section 2. For the analysis, and to provide some insight and basis for the comparison, a simple case study of a grocery shop is introduced and modelled using the different modelling techniques in section 3. In section 4, a framework to guide this comparison was developed and introduced together with the corresponding analysis of each technique. Some conclusions are drawn in section 5.

8.2 Modelling Techniques and Related Theories

8.2.1 Ontology charts

OC is the diagrammatic language used in Stamper's OS theory and the outcome of applying the method of SAM to an organizational problem. The underlying theory – OS theory – is based upon two simple assumptions:

1. There is no knowledge without a knower, and
2. His knowledge depends upon what he does.

These assumptions lead to the notion that any language for representing business, organisational, or social knowledge should have well-formed formulas (wffs) with the following syntactic structure: <knower-term> <behaviour-term> or <agent-term><action-term>. The behaviour-term relates to actions afforded together by the agent and the environment. Individual elements in the environment are seen from the perspective of the actions they make possible or afford for the agent and are referred as *affordances. Affordances* are the patterns of behaviour afforded by a particular element. For example, a cup may be seen as an affordance because it affords drinking, holding liquids, throwing it, (patterns of behaviour). Some affordances cannot exist without the existence of others; *swimming* is not possible without being *immersed in water*. In this case we have an ontological (or existential) dependency between affordances. Ontological dependency is the main relationship between affordances. Besides behaviour, which is afforded by the environment, there are also *norms* that guide agent's behaviour. These norms are social in nature, like laws, regulations, rules, which constrain or determine agent's actions. Norms are shared by people and can jointly form information fields (IF) leading to a new paradigm – the IF paradigm. An IF corresponds to a set of norms, shared by a specific social group such as a family, an organization, and a department that regulate their expected behaviour.

Agents and affordances are represented as nodes in OCs, while ontological dependencies (OD) are the lines connecting these nodes. Specialised OD includes generic/specific and whole/part relationships. Affordances can be substantive, representing here and now, or semiological, standing for other affordances. Agents may have roles in the scope of an OD. Time is also present in OCs – leftmost affordances must exist for the ones on the right side to exist.

A missing element from OCs, although an important concept in OS, is Norms that are represented in OS using a formal language. Although absent, they can be and are implicitly attached to affordances specifying, among other things, the start and finish of those affordances and the authority (ies) responsible for acknowledging its existence.

8.2.2 UML activity diagrams

The unified modelling language (UML) is "… a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems" (OMG 2003).

UML was adopted as a standard by the Object Management Group (OMG) in November 1997 and was the result of merging and adapting different notations from different methods and methodologies used for analysing and designing software systems, which applied the object-oriented paradigm.

UML became one of the most used modelling languages in the "software development world" and its flexibility, and extension capability allowed to be adapted and used in different areas such as business processes and information systems. UML defines nine diagram types (version 1.5 and before) which are described in Table 1.

In this work we will be interested in business processes and information system models, which exclude UML implementation diagrams. From the remaining diagrams the most commonly used for business modelling are activity diagrams, which enable the representation of the popular flowcharts and workflows. In addition, but not analysed here, case diagrams can be used to describe system functionalities and class diagrams to describe a static or structural view of different (business) entities and their relationships.

In this chapter we will be interested in activity diagrams as a complementary and standard diagram for our comparative purposes. Activity diagrams are used to represent activities, actions, and their sequence, sometimes presented and related to a responsible or performing actor that can be a human or a system. A course of action or sequence can be split in alternative routes by adding a decision element and a guard condition that will specify the route according to the (true) value of the condition. It is also possible to represent parallel actions using a *fork* at the start of the concurrent group of actions and a *join* element at the end. Objects associated or necessary to the actions and activities can be shown as attached notes.

Table 1. UML diagrams

Diagram type	Diagram name	Use
Structural (static)	class	To show classes (or classifiers) and their attributes and operations. Also relationships among classes are shown. Represents a static structural view of the entities and their relationships.
	object	To show a snapshot of the detailed state of a system, namely objects and data values at a point in time.
Behavioural (dynamic)	use case	To show the relationship among use cases within a system and their actors. Tell us the functionality of a system.
	sequence	To show an interaction or the exchange of messages among objects and/or actors. It provides a temporal view of the messages being exchanged.
	collaboration	To show collaborations containing a set of roles and the required collaboration relationships. To describe the realisation of an operation or a classifier.
	statechart	To describe the behaviour of instances of a model element such as a class instance or a specific operation. Includes possible sequences of states and actions in response to events.
	activity	To show a sequence of actions. It is a special case of a state diagram in which states are action states and transitions are automatic upon completion of the actions inside action states.
Physical (implementation)	component	To show dependencies among software components. It's a structural and logical view of software components and their relationships.
	deployment	To show the configuration of run-time processing elements and the software components, processes, and objects that execute them.

8.2.3 Role-activity diagrams (RAD)

RADs were introduced by Holt *et al.* (1983) and further developed by Ould (1995; 2005). We will refer to Ould's RADs in this comparison.

RADs constitute the main modelling language used by the Riva method (Ould 2005), which is a method for the elicitation, modelling, analysis, and design of organisational processes. In Riva we are interested in business processes understood as "a coherent set of actions carried out by a collaborating set of roles to achieve a goal" (Ould 2005, p. 32). RAD shows business processes personalised by one or more interacting roles. Roles are acted by one or more people – the actors which carry the responsibility for that role. Computer systems can also be modelled as roles. A role is represented as a shaded area containing a sequence of actions, including alternative and concurrent paths of actions, role states and role state descriptions, and initiators of actions (triggers). Actions are considered atomic and no further decomposition is allowed. An important aspect shown in a RAD is the interaction between roles, which represent explicitly the collaborations between roles. The start of a role can also be presented in a RAD.

8.2.4 DIPLAN plans

The DIPLAN language, described in Holt (1988), is the diagrammatic language used by the Theory of organized activity (TOA) (Holt 1997). This language was adapted from Petri Nets and permits simulation and action sequence analysis. TOA is based on "human" activities, which occur within every organisation or business system. Human action is the key element for the structuring and planning of all activity processes. An *action* in TOA corresponds to the unit of human effort, whereas *bodies* represent material or physical units. Every action is doubly performed by *organisational entities,* for example, a department, a president, a committee, and by a *person.* Actions and bodies are related by *involvement*: every action involves at least one body; every body is involved in at least one action. TOA defines different types of involvement between actions and bodies, namely creation, destruction, support, use, state change, and definition. An important aspect of bodies is that only bodies can have *states* which create alternative sets used for decision. According to this principle *information* consists only on these alternative sets used for decision. For example if someone decides to buy a pen only the elements such as price and writing colour *effectively* used in the decision constitutes information.

TOA is social in nature, any element or *unit* as called by the theory is acknowledge or identified by a *criterion* maintained by a community bound together by a shared organised activity – a *criterion* by which its members decide whether a given something is, or is not, a realization of the *unit*. The diagrams defined by DIPLAN are used to present plans describing organised activities. In DIPLAN the main elements are actions, bodies, and (involvement) relationships among them. Bodies represent physical objects either humans or materials, and are related to the spatial dimension, whereas actions are related to the time dimension and always connected to a human performer.

8.2.5 Dynamic essential modelling of organisations (DEMO) models

The DEMO methodology is based on the language action perspective, mainly inspired by the book of Winograd and Flores (1986) and based on James Austin's speech-act theory (1962), which was formalised in part by John Searle (1969). This book presents and proposes a design of computer systems based on a linguistic model of conversation for action. A language-act may be explained as the utterance of a sentence seen as an action. In this case it is called a performative utterance having a general form given here through this rough definition: When we say something (a locutionary act), with the intention or the effect to change the world (or acting on the world) in some way, we are performing an illocutionary act, being those changes perlocutionary acts.

DEMO defines the elements of a communicative act, with an illocutionary part and a propositional part that is represented formally by the OER[1] notation:

<locutor> : <illocution> : <addressee> : <fact> : <time-for –completion>

Connecting these elements forms the DEMO basic building block of every business system – the business transaction. A business transaction is composed by an order phase, an execution phase, and a result phase. The first and last phases are seen as performative conversations and are used to reach an agreement respectively, on the request and on its successful result. The middle phase is the necessary (objective) action associated with the request. DEMO sees a business system as "a coherent structure of transactions" and proposes six different models each one associated with a correspondent diagram. Table 2 gives a brief overview of these diagrams.

In order to keep this analysis simple and short, we will use only interaction diagrams which identify the main elements of DEMO approach, namely: actors and transactions. The other models go further in the analysis by giving detailed description of transactions such as their phases and sequences, structure, states involved, and information facts.

[1] OER stands for Order Phase, Execution Phase and Result Phase.

Table 2. DEMO diagrams

Diagram name	Use
Interaction	To show the different transaction types (business transactions at the type level), initiating and executing actors, and the system boundary.
Business process	To show the transaction phases, and their causal and conditional or optional relationships providing a time aware view of business transactions.
Transaction process	To show the structure of transactional communication by describing the possible communicative actions and transaction states in a business transaction process.
Action	To show action rules through procedures (a kind of flow-chart) that is applied to every distinct non-terminal state of each transaction type.
Fact	To specify in a precise and complete manner the information space of an organisation under consideration.
Interstriction	To show informative conversations, information banks, and actors.

8.3 The Grocery Shop Case Study

8.3.1 Case study description

For our analysis we decide to use a simple case study taken from a common everyday living situation. This case study intends to be as simple as possible without any previous connection with IT. It portrays a grocery shop, as follows:

> *A grocery shop sells fruit to its clients. A client enters the shop and chooses the fruit he wants to buy. After choosing the fruit the client gives it to the employee, who weights it and calculates its price based on the weight. The client pays the value asked by the employee.*

This text should be understood as a possible description of his/her business made by the grocery shop owner using his/her own words and terms, and probably missing any details he/she did not feel necessary to mention.

8.3.2 Case study model implementation

For illustrative purposes and as a concrete basis for our comparison this case study was modelled using the techniques described before, the results are presented in Figs. 1 and 2.

In order to show the adequacy and application of each modelling technique we think it will be useful to report a few difficulties we felt creating these models as follows:

Ontology Charts

"Choose" or "weigh" *look like* actions and are difficult to relate to the affordance concept. As *affordances* they should enable other actions and therefore stay valid or exist during the existence of the actions they afford. But, as actions they are transient and they cease to exist after being done. In fact they are represented using action terms but they mean "after" action states.

This is not shown in the OC and it would go against the rule of only two ontological antecedents for each affordance.

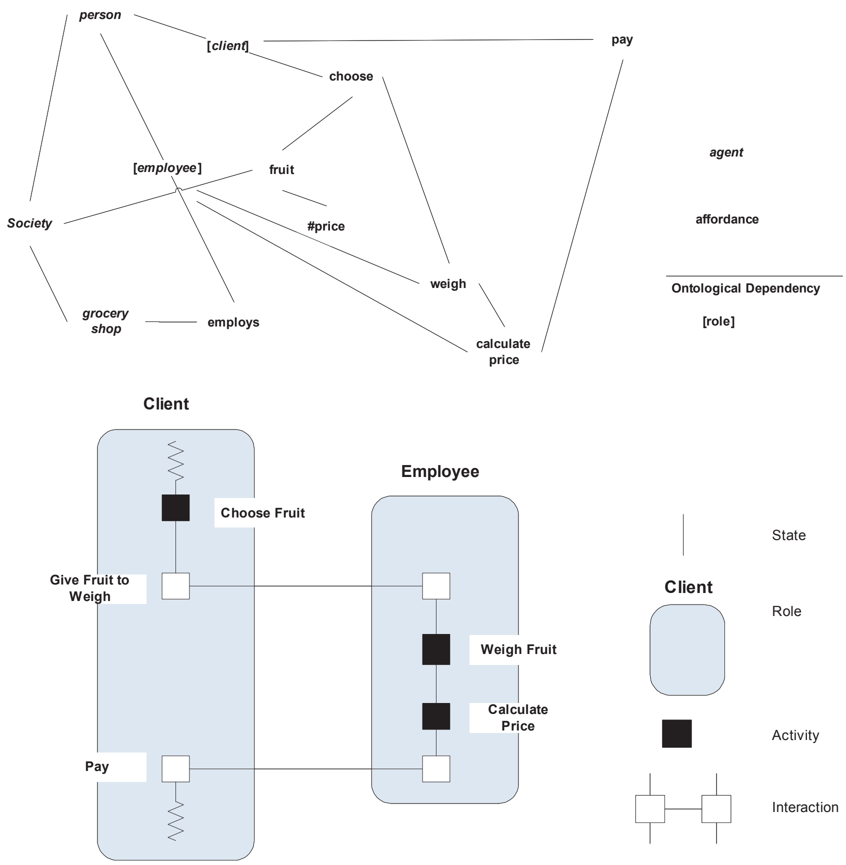

Fig. 1. Case study model diagrams – OC (*upper*) and RAD diagrams (*lower*)

Role Activity Diagrams

The fruit object is mentioned but has no place in the diagram.

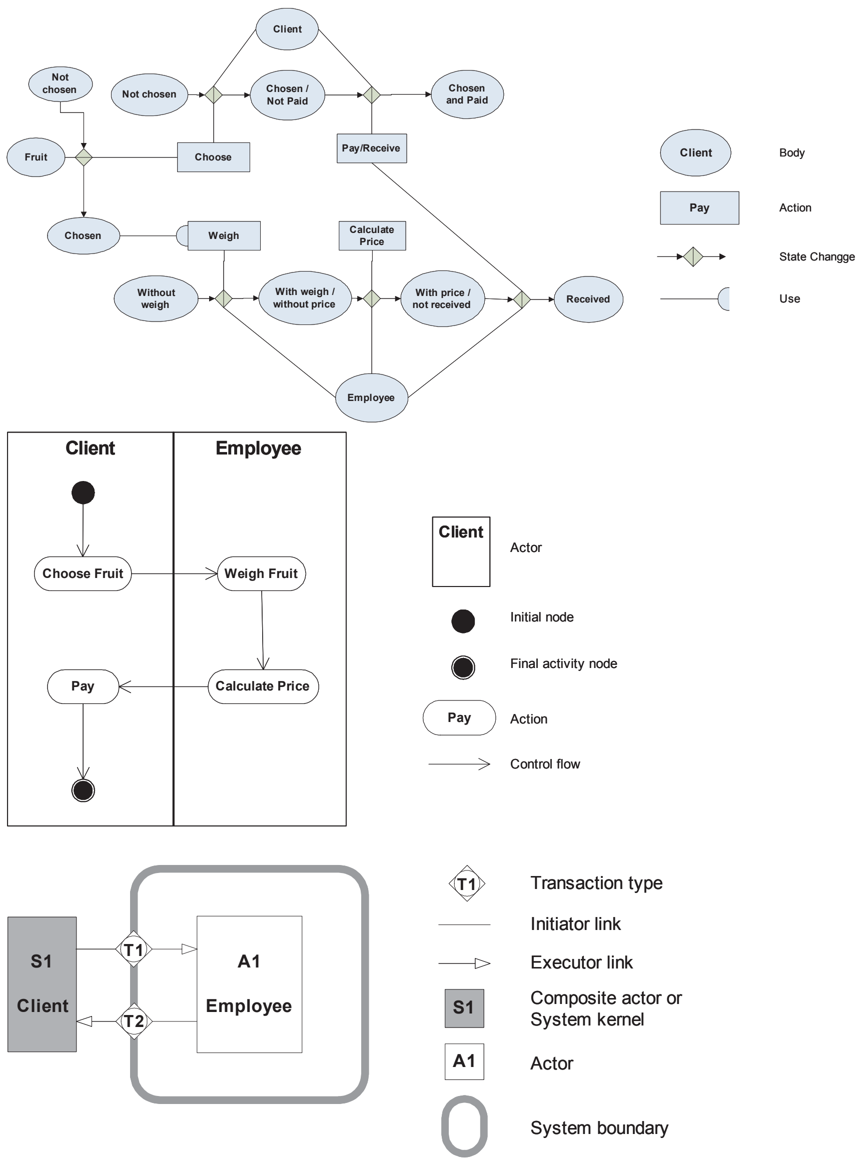

Fig. 2. Case study model diagrams – DIPLAN (*upper*), UML Activity diagram (*middle*) and DEMO interaction diagram (*lower*)

DIPLAN

The resulting diagram is too big and complex for this simple case study.

We could not clearly show action sequence particularly relating to the "choose" and "weigh" actions.

It was necessary to mix different states within the same body. For example "chosen" and "not paid" within the client actor.

DEMO Interaction Diagram

T1 and T2 in the diagram represent, respectively, the "request to take the fruit" business transaction and the "payment request" business transaction. The meaning of these *codes* has no place inside the diagram and needs to be explained.

The action of *choose the fruit* is not possible to represent naturally in the diagram because it is not initiated by a speech act – there is no order phase.

As in RADs the fruit object has no place in the diagram.

UML Activity Diagram

The pay action is only shown inside the area of the actor "client". If modelled as a receive action it would appear only inside the "employee" area. It could be depicted over the division line of both actors but in that case we would need to change the action name, *pay* term would not be possible.

The fruit object was not shown but could have been included as a note.

8.4 Comparative Analysis

8.4.1 A comparative framework – the "common questions framework"

Our comparison will focus on what is expressed by the different modelling techniques described before. Practical use and usefulness of the models and diagrams will not be considered. Therefore, in order to identify and examine what is being expressed it is important to define the comparison items and dimensions. Typical dimensions or views commonly used in comparisons are the organisational, functional, behavioural, temporal, contextual, and motivational dimensions. Although there is no agreement on this subject we can relate these dimensions to the traditional who, what, how, when, where, and why questions. But, are these good comparison dimensions? The answer will depend on our goals and these will depend on our roles as business designers, systems designers, system programmers, users. Because we are interested ultimately in IT support for organisations we will take into account two general roles or groups of people, namely business/organisation people and IT people. These two groups often found in information systems areas represent two different views of information systems, a business oriented perspective, and a technological oriented perspective. Aligning these views, the business-ICT alignment, is many times claimed as a goal for achieving a better technological support for business systems. Because we are still seeking opportunities for business IT support our analysis will be based on the business people view. Their view will give us the necessary hints for the IT support of business and organisations.

What about goals? Usually our goals are related to understand, change, improve, manage, control, plan, or even start a business or organisation. Again a traditional division can be established using the analysis and design perspectives. If the intention is to first understand business and organisations we will be performing analysis. Our concerns will be related to present and past situations, this will be mainly "understand businesses". Otherwise our goals will be directed to change the actual situation and this will result in the "design" of a new one. In this case our concern is about future situations including planning, improvement, redesign, etc. In this comparison we will focus on analysis or business descriptions, although any description can be used as a design for a future organisation.

Getting back to the stated dimensions and in order to provide a clear separation between them and to extend our comparison framework we would like to explicitly state the meaning and the view provided by each of them and to add two other important dimensions: a conceptual and a resource dimension. These dimension will be related to the which and the with questions. A more detailed description of each of the proposed dimensions follows:

- *Organizational (who)* – To identify people involved and how they are related or grouped. We will be interested in items like departments, teams, and projects. Typical organisational charts will be a good example. Also social power structures like an organisational hierarchy can de depicted from this dimension.
- *Functional (what)* – Related to organisational or people functionalities. Used to show what actions or activities are provided by a system or organization. Applied to organisations or organisational units will tell us what actions these *systems* would perform. People can be shown here by their roles, skills or knowledge within a functional perspective.
- *Behavioural (how)* – To show detailed functionality. It will provide further detail of functionalities such as descriptions of actions

sequence, resources used, and people involved. Examples could be processes, and procedures.
- *Temporal (when)* – To show task allocation and its time duration. Temporality will be associated as well to states and events. Typical examples depicting this dimension are Gantt Charts, Pert Charts, Petri Nets and Statecharts.
- *Contextual (where)* – Related to place or location for action and resources. It is a view on where actions take place or where resources are kept. Context can also be applied to people and in this case we will be referring to social context such as cultural, political, and emotional.
- *Motivational (why)* – To address motivation for actions, people or organizations, could include goals, intentions, interests, purposes, objectives, and aims.
- *Resource (with)* – Related to means used by people or organisations. Included are people as actions performers, materials, tools as action enablers or supporters and documents as information providers.
- *Conceptual (which)* – To identify important business concepts and their relations. A structural view of business elements to give us a general vision.

All these dimensions are related to common business and organisational aspects but there are some other concepts not addressed. In this case this is because it is difficult to express them using diagrammatic languages, examples can be risk, value, cost, culture, and ethics. Although some of them are possible to represent as model elements in diagrams such as conceptual diagrams, they are just suitable for being expressed as texts and possibly included as notes in diagrams or documents attached to the diagrams. A last important element not considered as well is business rules, or rules in general such as regulations, laws, norms, and procedures. These concepts are suitable to be expressed mainly through textual languages including formal or semi-formal textual languages. Again their use in diagrams is mainly restricted to notes or texts attached to other model elements. An exception to this textual representation includes just some business rules that can be shown using actions and then presented using flowcharts or similar diagrams.

8.4.2 General remarks

Our framework was applied to the techniques described in section 2 and the result is shown in Table 3. Looking at the results presented in this table we note the absence of elements to represent the contextual dimension, and also the motivational dimension. This happens due to the problem referred to earlier related to the difficulties in showing diagrammatically these aspects.

Another note should be added to the case of the temporal dimension. The concrete time aspects relating tasks, their time duration, and the related performing persons typically addressed by the common Gantt and Pert Charts does not have an equivalent in any of the modelling languages analysed. Connecting or relating people to action using roles or other concepts seems to be an essential characteristic of any organisation system.

A last general remark concerns the organisational power hierarchy commonly expressed using organisational charts; again none of the present techniques show us this view.

Table 3. Techniques comparison table

	OC	DIPLAN	RAD	DEMO	UML
Organisational (who)	agent (authority)	(Human) body organisational entity	role	actor	actor
Functional (what)	—	activity	process and roles	business transaction	use case
Behavioural (how)	—	actions	action interaction	business transaction	action
Temporal (when)	ontological dependency	state	state event	state transition	state event
Contextual (where)	—	—	—	—	—
Motivational (why)	—	—	state description (goal)	—	—
Resource (with)	affordances	body	—	information bank, facts	object
Conceptual (which)	affordances	—	—	—	classifier

8.4.3 Ontology charts analysis

A simple look at the table shows us that OC covers a small part of the questions posed in our framework. In fact, and given its characteristics, it is not suitable for describing operations, what is done and how it is done. Affordances represent patterns of (possible) behaviour, but this behaviour is not explicitly represented. Another aspect is that ontological dependency is based on the idea that affordances (or the behaviour afforded by it) are not possible without the existence of other affordances or agents for realizing the associated behaviour and this is shown using words, such as verbs as affordances that are related to actions. For example in the case study we find that "choose" is an affordance that cannot exist without a "client" to choose and the "fruit" to be chosen. This can be a misleading aspect because we will be interested in existence and this is associated to *existence states*. To correct the diagram from this point of view what should be represented would be "choice made" or "fruit chosen" which is a *state*. Weighing does not depend on "choose" (that is a transient state – a time-limited action) but on the existence of a choice. This idea will give us a picture of ontology charts as an *existence state chart* where only existential states are represented. This is something that we cannot find in other approaches to ISD and it could be very useful. Traditional state charts represent transient states that are reached at some point and are left through a transition. This is captured using the temporal dimension, although in this dimension most techniques do not show us exactly when things happen. Regarding the conceptual dimension we find that the OC elements can in fact be used to represent conceptual notions if we choose to represent concepts as affordances and establish their existential relationships. The same can be said to the representation of resources using the resource dimension. Looking at the common use of OCs we experienced a mixed representation of concepts, resources, and other affordances with a particular distinction of substantive and semiological affordances to acknowledge the classification between actual (possible) behaviour and future behaviour expressed by actual affordances (i.e. a plane ticket standing for a plane trip).

Finally an important aspect of the OS approach is the social issues which are depicted in OCs through agents and some (conceptual) affordances. The hidden norm elements not shown in the diagram do not allow for an explicit view of these concepts that enable us to state conditions for the existence of affordances, and to give information about the agents responsible for recognizing that existence (the *authorities*).

8.4.4 DIPLAN analysis

As shown by the Table 3 this technique also presents a very limited scope by giving just a view of the actions and (body) states and their sequential order within the organisational activities. This is much like the behavioural dimension of the comparison framework. An overall picture of the organisation functionalities, the functional dimension, is only addressed indirectly by looking at each diagram – the activity – as a whole. Anyway, also the temporal dimension is represented through the expression of body states and the action transitions enabling this model to be used in simulations and to have a practical use that is similar to Petri Nets. Besides this limited application a note should be made of an important aspect, which is the explicit representation of persons responsible for each action. This enables us to establish responsibilities and commitments and to have a complete expression of the OS's agent-behaviour assumption.

The organisational dimension is also acknowledged by representing organisational entities but an (organisational) structure of these elements is not presented.

8.4.5 RAD analysis

RAD is the only diagram that addresses the motivational dimension by explicitly including goals as state descriptions. This could be helpful to understand and do the assessment and control of business processes. As other modelling techniques presented here the main dimension addressed is the behavioural dimension. The functional dimension is also represented looking at the roles seen as processes without acknowledging the detail. Interestingly the organisational dimension can be seen or extracted by looking at the roles and their interaction. This permits to see what roles exist and the (interaction) relationships established among them but not other kinds of relations such as a power hierarchy usually depicted using organisational charts. As DIPLANs, RAD also allows us to simulate business processes thus addressing the temporal dimension. The approach is similar to Petri Nets using a token that will navigate each state (represented as a line in a RAD) and enabling to show the current state of a role process. Another consideration to RAD is that this kind of diagram does not give much attention to the resource dimension from a material, tool, or document perspective. These elements are not shown in the diagrams. A final remark about another diagram of the Riva method – the process architecture diagram – not analysed here, shows the process architecture.

8.4.6 DEMO diagrams analysis

Demo modelling as organisational semiotics uses a different paradigm to information system analysis in this case based on human communication and networks of commitments. Using this approach leads to a representation based on business transactions that model the communicational aspects of organisations. The identification of actors from the organisational dimension is automatically done with the identification of initiating and intervening actors of business transactions. Organisational charts with other kind of relationships besides the communicational interactions are not represented.

Behavioural and functional dimensions also see business transactions within a behavioural view using interaction diagrams and within a detailed functional view using business process diagrams. Other diagrams, including business process diagrams focus on detailed aspects of business transactions, including structure, causal and conditional, or optional action transitions, action states, and action rules. These aspects enable also to represent the temporal view as in the case of RAD and DIPLAN diagrams analysed before. It is important to mention that these functional, behavioural, and temporal dimensions are partially represented: only communicational actions are detailed and other types of actions, called performative, which are related to objective action are only presented; no functional or temporal representation on how are they made is given. Looking at the case study an action like choosing the fruit is not presented, in this case the first business transaction T1 is related to the request to take the chosen fruit.

Due to the focus on communication also resources are not completely addressed as they are seen from an objective perspective, even though this leaves informational resources untouched and these are considered for the facts perspective of business transactions or communicative acts.

8.4.7 UML analysis

In spite of its software specification orientation, UML diagrams provide the most complete set of diagrams in order to cover the proposed dimensions. The functional dimension is represented using use cases and actors and has a specific use case diagram for its representation. The same can be said with the temporal dimension and statechart (and activity) diagrams. The behavioural dimension is covered with activity diagrams and sequence diagrams. Giving its metalevel definition it is possible to change the semantics of the elements using the most powerful UML extension mechanism, the *stereotype.* Departing from the *classifier* metalevel element is possible to instantiate it in the form of a class or a use case. This feature allows representing concepts as conceptual elements (adapted classifiers) and their relationships using mainly class diagrams. This permits the representation of business concepts from the conceptual dimension.

Although there are several dimensions in the framework addressed by the UML set of diagrams, there are some problems related to the representation of human aspects. One common problem is to acknowledge human responsibility for actions; a human is not usually associated with a specific action as in the case of a DIPLAN action, where he must always be present. Human interaction in the form of communication or any other activity is not represented as well. Many of these factors can be overcome by defining a specific UML extension to handle this but with the remaining problem of the metalevel structural assumptions implicit for each UML model element.

8.5 Conclusions and Future Work

In this chapter we tried to understand and situate OC within different other diagrammatic techniques by analysing and comparing it. OC presents a different view not addressed by other techniques which can be helpful to the design of stable information systems. Its *existential state oriented* characteristic should be considered important for element dependency analysis. This dependence is responsible for many of the problems experienced with organisational change. A question still remains however about the scope of these diagrams: Should not OS provide and give more attention to other aspects of business such as operationalisation as approached by the functional and behavioural dimensions? Within OS it is difficult to explain how something will be done. Norms regulate the actions and describe the expected pattern of behaviour but the concretisation of this behaviour is left out.

We are particularly interested in understanding the application scope of OCs. Based on the comparative analysis above, we identified the essential aspects of OCs that differentiate this approach from others. We strive to make the advantages provided by this methodology available to a broader Soft Systems Engineering community. As future work we plan to create an UML extension in order to express OCs with UML, so we can have a wider audience for OS concepts.

References

Austin JL (1962) How to Do Things with Words, 2nd edn, Harvard University Press, Cambridge, MA

Booch G, Jackobson I, Rumbaugh J (1998) Unified Modeling Language User Guide, Addison-Wesley, Reading, MA

Cordeiro J, Filipe J (2003) Language Action Perspective, Organizational Semiotics and the Theory of Organized Activity – A Comparison, Proceedings of the workshop DEMO, Tilburg, The Netherlands

Holt A, Ramsey HR, Grimes JD (1983) Coordination system technology as the basis for a programming environment. Electrical Communication, 57(4), pp. 307–314

Holt AW (1988) Diplans: a new language for the study and implementation of coordination. In ACM Transactions on Information Systems (TOIS), 6(2), pp. 109–125

Holt A (1997) Organized Activity and Its Support by Computer, Kluwer Academic, Dordrecht, The Netherlands

Liu K (2000) Semiotics in Information Systems Engineering, Cambridge University Press, Cambridge, UK

OMG (2004) Unified Modeling Language Specification Version 1.5. Available at: http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML (June 2005)

Ould MA (1995) Business Processes – Modelling and Analysis for Re-Engineering and Improvement, Wiley, Chichester, UK

Ould MA (2005) Business Process Management – A Rigorous Approach, The British Computer Society, Wiltshire, UK

Searle JR (1969) Speech Acts, Cambridge University Press, London

Stamper R (1996) Signs, norms, and information systems. In: Holmqvist B *et al.* (eds.), Signs of Work, Walter de Gruyter, Berlin

Stamper R (1973) Information in Business and Administrative Systems, Wiley, New York

Stamper R (2000) Information systems as a social science: an alternative to the FRISCO formalism. In: Falkenberg, E, Lyytinen, K, and Verrijn-Stuart (eds.), Information System Concepts: An Integrated Discipline Emerging, Kluwer Academic, Massachusetts, pp. 1–51

Van Reijswoud VE, Dietz JLG (1999) DEMO Modeling Handbook, Available: http://www.demo.nl (September 2005)

Winograd T, Flores F (1986) Understanding Computers and Cognition, Ablex Publishing Corporation, Norwood, NJ

Chapter 9
The Separation of Data and Information in Database Systems under an Organisational Semiotics Framework

X. Wang, J. Feng

Database Research Group
School of Computing
University of Paisley

Abstract

Some significant problems in database systems research, such as query answering capability, connection traps, lossless transformation, and normalisation are difficult to explain, answer, solve, or explore further within the current context where data and information are fused, or even taken as the same thing. It would appear that a fresh and new perspective that distinguishes between data and information, and takes data as a type of sign, which bear (carry) information, is beneficial. The ideas of organisational semiotics (OS) enable this. We will discuss the reasons for separating data and information, and for investigating the relationship between the two. We will look at how and why the contemporary, seemingly muddled view, on the relationship between data and information might have hampered the progress on a number of database research issues. We conclude that looking at data and information within an OS framework sheds light on these issues and helps investigate them with a sound theoretical ground.

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 165–176.

9.1 Introduction

The concepts of data and information are fundamental in database systems research. Although many researchers have been working on them, it would seem that unified definitions on data and information are unlikely to be reached in the near future.

Traditionally data is taken as raw material whilst information is generated from data after being summarised or analysed. One of the typical definitions is: "…data consists of the raw facts and figures that are processed into information. Information is summarised data or otherwise manipulated (processed) data" [13]. In Elmasri and Navathe [8], a database is explained as a collection of related data and is specified as facts that can be recorded and have implicit meaning. Checkland [3] proposes "information equals data plus meaning". "Information is data that has value. Informational value depends on context. Until it is placed in an appropriate context, data is not information, and once it ceases to be in that context it ceases to be information". [4]. Mingers [20] argues that "meaning is created from the information carried by signs. The consequences are that information is objective, but ultimately inaccessible to humans, who exclusively inhabit a world of meaning. Meaning is essentially intersubjective — that is, it is based on a shared consensual understanding."

Database systems are vehicles of storing and providing information [1]. Even through the above brief citations of relevant literatures, we find that most of the time information is fused with data. What you can see is what you can get in a database. One prevailing view is that data which has its meaning equals information in the context of database systems. And this view has been adopted as an implicit assumption in some database research, one of which is notably the Relative Information Capacity theory [14]. This theory reveals the fact that data instances of schemata are taken as "information" in databases and the "information capacity" of a schema is the capacity of accommodating instances into it. We argue that instances cannot be simply taken as the information in databases and that it would be a flawed theory if the confusion of data and information exists.

The rest of the chapter is structured as follows. We briefly state the theoretical foundations of this study and discuss the possibility of drawing from them to address the problem at hand in section 2. We give some analysis on the need of separating data from information in section 3. We enumerate some questions in database research, which have not been answered very well due to this confusion of data and information in section 4. The benefits of clarifying such basic and fundamental issues are also discussed in that section. And finally the conclusion of this work and our work on the "information bearing capability" of data schemas is briefly described in section 5.

9.2 Theoretical Foundations

Following one of the basic ideas of OS, namely signs and their properties presented in [29], a formal information system can be seen as a system of signs. This enables the distinction between data in the system and information that data represents. Information flow and information transmission cannot happen unless there is some bearer/carrier for the information, i.e. the existence of signs defined broadly.

Although OS falls in a subjective dimension, it argues that sign is a commodity other than information. According to [27], [28], and [29], information can be equated to properties of signs. Various definitions of information correspond to properties of signs on various semiotic levels. In OS, the very notion of sign is defined as something that stands for something else, and this is taken as given. With the *Semantic Theory of Information* [7] which was originally published in 1981 and the information flow theory [2], we can explore why this is the case. Because of the existence of information flow from A to B, B has the potential to be a sign that refers to A. We can also say that the information content of B includes something about A, and B is a bearer (carrier) of some information about A. On the one hand this is how a sign can ever possibly be a sign at all. On the other hand, OS enables us to look at the properties of information bearers systematically on all the semiotic levels. OS and information theories therefore supplement each other and possibly help develop each other further.

We try to propose an ostensive way of understanding data and information, following which no such confusions should appear, and some of the flaws in current relevant researches can be explained. Our attempt to separate data and information in database systems might bring new concepts and ideas to database systems and so a new perspective might appear for databases.

9.3 Why Separating Information from Data?

The previous section gives theoretical reasons for separating information from data. We will now examine more specific and practical reasons for such separation within the context of database systems.

9.3.1 Instances do not necessarily carry the information that results in their "types"

Data stored in databases are schemas and instances of these. It would seem plausible to think that a schema represents the "type" level information, and instances of the schema represent the information on the "token" level. In such a sense, it might seem reasonable and sound to take that data as the information in the database. However, we want to argue why data is not information exactly from this perspective.

All the related things on the type level, i.e. the structure, constraints, and legitimate operations [30] on the data, of the schema contribute to the meaning of the data that is stored in the database. So schema itself can be seen as "concepts" in the sense of Dretske ([7], p.214). Schema also captures the relationships between concepts, which are in a broad sense also concepts themselves.

Calvanese *et al*. [5] argue that a schema only determines necessary condition(s) for data to be qualified to instantiate the schema, but not sufficient condition(s). According to Dretske, instances of a concept inherit everything from the concept and it has the capability of "giving meaning" to its instances. But schema cannot guarantee that the instances that are put into the schema are right or true. The formation of a concept needs the right information while something to be deemed as an instance of the concept does not need that piece of right information.

For example, in cartography, a blue wiggly lined area on a map represents a lake. If a map maker mistakenly put a lake symbol on a map where there is no such lake in reality, the map can misrepresent the geography of an area only insofar as its elements are understood to have a meaning independent of their success in carrying information on any given occasion ([7], p.192). "A symbol token fails to carry the information that, in virtue of the type of which it is a token, it is its job to convey." ([7], p.193)

Due to these special characteristics of types and tokens, or concepts and instantiations, and the complex relationships between them, it is deemed that problems will occur if simply taking data as information. Instances do not always exactly inherit the information content of its concept. In this sense, instances are not reliable to be taken as information.

9.3.2 Information content vs. literal/conventional meaning

Ultimately it is the information content of a database that the researcher and the user of the database are interested in and this is what really matters. The "information content" of a conceptual data schema has been recognised to be "difficult to define and measure" [1] and [18]. It would seem that as a result of this, instead of digging into the complexity, and sometimes rather deep philosophical issues, which would be required for studying information content, and due to its simplicity and straightforwardness, many database researchers have concentrated their effort on data and their meaning.

Taking one step back however, we see that the meaning of data in a database is taken as information (content) that the data express. Literal or conventional meaning of data is probably the most convenient to obtain. As a result, literal or conventional meaning of data is granted the status of information (content). We argue that the meaning of data is not necessarily their information content, i.e. what the data represent.

Following the ideas of OS, an information system is a system of signs. Signs are the bearers of information. To represent a piece of information, there could be various ways and therefore various bearers for the same piece of information. As long as the system offers some means to infer the information content from the bearer, all these bearers are valid and practically feasible. So we say that for a data construct to be capable of representing a piece of information, the information content of the data construct, when it is considered in isolation, must include the information content of that piece of information. The simplest case is that the literal or conventional meaning of the data construct is part of its information content and they represent what are required. It seems too restrictive or unnecessary, and theoretically unsound for a database to impose the constraint that for some data to represent a piece of information, the information content of the data has to be the literal or conventional meaning of the data.

9.3.3 An analysis of the RIC theory

As mentioned previously in the introduction, the relative information capacity is a typical example in database research where data (instances) are taken as information. We will analyse why this view is problematic and only valid within a rather narrow context.

The notion of information capacity preserving is originated from the relative information capacity (RIC) theory [14]. It is concerned with information preserving mapping, dominance, and equivalence for simple conceptual data schemas and it is used as a correctness measure for schema transformation with no information loss.

Four progressively less restrictive dominances were proposed. These dominances and equivalences are based on the existence of abstract functions of some particular type between the instances of a pair of schemas.

But it is not clear how to reason about the existence of such abstract functions, which are crucial for RIC. Miller *et al.* [21], [22], [23], and [24], redefine the notions of absolute and internal dominance among the four and put forward the schema intension graph (SIG) data model with a view to enable reasoning about the existence of the abstract functions. The literature would seem to show that the RIC theory and the SIG model are widely accepted and used [12], [16], [17], [19], and [32].

We have studied the SIG formalism and have, we believe, a number of significant findings. The basic idea of information capacity preserving can be stated that if every valid instance of a schema can be represented as a valid instance of another schema, which can be recovered from the latter, then the latter schema is said to dominate the former and have a greater information capacity than the former. For example, if two data schemas S1 and S2 are the same except that S2 has fewer constraints than S1, then S2 would be deemed to have a greater information capacity than S1 simply because all valid instances of S1 can be accommodated in S2. It sounds plausible. However, if we take a closer look, it can be seen that the fewer constraints one schema has, the less specific the instances would be, and therefore the less informative the instances become. For example, a relationship with a "many to one" cardinality ratio is seen, following the RIC theory, dominating one with a "one to one" cardinality ratio, because any instance of the latter can be stored as instances in the former. That is, the former has a greater information capacity than the latter, i.e. the information capacity of the former includes that of the latter. However, an instance of the former is less specific than that of the latter, and therefore less informative. That a student is tutored by a professor in a "one to one" session is more specific and therefore contains more information than when a student is tutored by a professor in a session of "many to one", which includes "one to one". This is because the latter involves more uncertainty (more possibilities) than the former.

Thus we observe that the basic ideas behind RIC and SIG are instance-centric and inappropriately taking data instances as the entire information that a data schema can provide. We argue that this viewpoint is questionable, unnecessarily restricting and it only makes sense when what is meant to be a valid instance is narrowly defined. One example of a valid instance is "a student is tutored by a professor" without considering the constraint of the cardinality ratio as mentioned above.

In conclusion then, through the above analysis, we argue that simply taking data as information in databases is questionable, inappropriate, imprecise, narrow-minded, and dated on some occasions.

9.4 Separating Data and Information Should Help Further Database Research

It would seem that due to lack of distinction between data and information, many researchers have been working on schemas and their instances exclusively. But problems such as the correctness of methodologies or techniques cannot be explained satisfactorily. Based on the above discussion, in this section we examine a number of issues in database research and show how separating data and information might help further investigate them, including to better explain these problems or phenomena in databases.

9.4.1 Connection traps

The term of connection traps in entity-relationship (ER) schemas seems to be first mentioned by Howe [19] in 1989. Later connection traps including chasm trap, fan trap, and "Y" trap were further studied in [9] in more depth. Howe takes the misinterpretation of the meaning of certain relationships as the reason for connection traps. Codd [6] mentions the "plurality of joins", which can be taken as an explanation of such phenomena although he himself did not bring forward and study these kinds of problem explicitly. It would appear strange that such problems have not caught much attention of other researchers.

In our view then, connections in an ER schema become problematic only because they are used to represent certain information that they cannot carry. More precisely, there are two types of situations. One is where there can be no data in the form of paths in an ER schema that represents a particular piece of information, and this is not recognised. The other is where there are data level connections (i.e. paths) that could represent a particular piece of information if they were distinguishable and yet they cannot be distinguished from the other instances.

For example, the problematic path in Fig. 1 cannot capture "which full-time lecturer is involved in the teaching of which courses run by the department", as there is a fan trap in the path. But the path does have some topological connections that represent the correct information, even if you could select them out, you just cannot tell which ones represent the correct information. So in relation to this information, the path is an instance of the second situation mentioned above.

Solutions to such problems include classifying paths in terms of the information content and the distinguishability of a path. For any given query, any given path is one of seven classes of paths. For details about connection traps and possibly solutions, the reader is referred to [10].

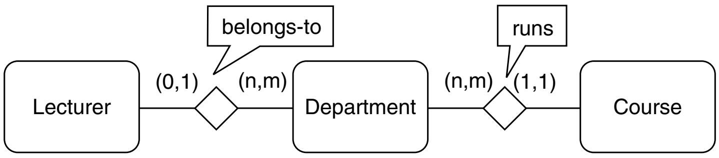

Fig. 1. A path in relation to "which lecturer is involved in the teaching of which courses"

9.4.2 Query answering capability

Currently research on query answering capability is based on direct match of predicate expressions with objects in a database. Objects that are involved in a query normally have to be pointed out explicitly. It would seem therefore that such a perspective will not result in a complete query answering capability. Following Dretske [7], the semantic content of an information-bearing structure is the outermost informational shell, and it is the piece of information in which all other information carried by the structure is nested. There is an example (see Fig. 2. below) in ([7], p.178), which illustrates the concept of semantic content. If a signal S carries the information that t is a square, but carries no more specific information about t (red square, blue square, big square, little square, etc.), then S carries the information that t is a square and it is its semantic content. The information that t is a parallelogram, or a rectangle, or a quadrilateral etc. is nested in the semantic content that t is a square.

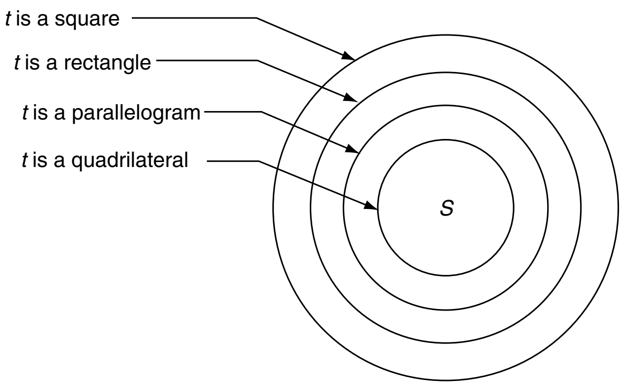

Fig. 2. An example of semantic content of an information-bearing structure and its information nesting

If the semantic content of a data construct can be described explicitly and certain reasoning mechanisms are available, the information nested in such a semantic content can then be reasoned about. That is to say, exact matching of predicate expressions with query objects is not necessarily required, and query answering capability should not be limited within such exact matching either. This extends the boundary of such a capability.

9.4.3 Correctness criteria for lossless transformation/translation

As we mentioned in section 3, RIC is a widely accepted theoretical correctness measure for schema transformation. It is an instance-centric theory. Although there are other approaches [18], [25], and [26], "instance dominance" is still a dominating guideline. We argue that looking at instance dominance alone is not adequate for discussing the information capacity of a data schema, not to mention that such an approach only applies to the scenario where the schemas are modelling the "same data" [14]. Even when modelling the same data, which could be taken as modelling a same application domain and therefore the same information requirements, there is normally more than one bearer, which would normally have different data instances in a database.

If we can develop an approach, from the perspective of "data bear information", on how to measure information loss [15] and [31] and information containment, it will be applicable to the transformation from a conceptual model to a data model, and from information requirements to conceptual schemas. These will possibly lead to proper correctness measures for any information systems.

9.4.4 Normalisation

A well known criterion for normalisation is attribute preserving, functional dependence preserving, and lossless join. Normalisation is focused on avoiding anomalies on the operational level, namely insertion, deletion, and updating operations. In fact, a normalisation process is that of schema transformation. And schema transformation should be information bearing capacity preserving. New measures of normalisations can therefore be developed with the perspective of "data bear information". To this end, we can look at whether the aforementioned criterion is equivalent to that of RIC; if so, we can then follow the criteria mentioned in the previous subsection, which will hopefully be worked out in the near future.

9.4.5 Summary

To summarise, the current seemingly dominant view that confuses data with information in database systems research might have hampered further exploration of the above enumerated problems. And using the theories of information flow, such as Drestke's semantic theory of information [7] and the information channel theory [2], within the framework of OS may well help further research. Using these theories, we find that for a data construct (a bearer) to represent a semantic relation (a source), the equivocation that is the information generated at the source which is not carried by the bearer to the user of the information (i.e. the receiver), should be zero. Moreover, many fundamental issues such as information content, information quantity, and the granularity of data constructs can also be studied.

9.5 Conclusions and Further Work

Data as the information bearer in a database should not be simply taken as information. Data is a kind of sign that carries information. The literal or conventional meaning of data is not necessarily equal to the information it carries. A special situation is where all the data in a database are true facts and there are no connection traps, i.e. misinterpretations of the capability of data carrying information. This seems to be a tacit assumption and sometimes unrecognised assumption for databases. The situation where some data in the database is not true is omitted, which is not reasonable or realistic. More seriously, this is unsound theoretically. The concepts of "information content" and reasoning about information nesting should further research on databases.

References

1. Batini C, Ceri S, Navathe SB (1992) Conceptual database design: an entity-relationship approach. Benjamin-Cummings Publishing, Redwood City, CA
2. Barwise J, Seligman J (1997) Information Flow: The Logic of Distributed Systems, Cambridge University Press, New York
3. Checkland P (1981) Systems Thinking, Systems Practice, Wiley, Chichester
4. Clarke R (2005) Fundamentals of 'information systems'. Retrieved from: http://www.anu.edu.au/people/Roger.Clarke/SOS/ISFundas.html, 6th April

5. Calvanese D, Lenzerini M, Nardi D (1998) Description logics for conceptual data modeling. In: Chomicki J, Saake G (eds.), Logics for Databases and Information Systems, Kluwer Academic Publishers, Norwell, MA, pp. 229–264
6. Codd EF (1970) A relational model of data for large shared data banks. Communications of the ACM, 13(6), pp. 377–387
7. Dretske FI (1999) Knowledge and the Flow of Information. Basil Blackwell, Oxford
8. Elmasri R, Navathe S (1994) Fundamentals of Database Systems, 2nd edn, Benjamin-Cummings Publishing, Redwood City, CA
9. Feng J (1999) An analysis of "Fan-traps" in EER modeling by using a set of "info concepts". In: Castell A *et al.* (eds.), Synergy Matters. Kluwer Academic/Plenum, New York, pp. 541–546
10. Feng J, Crowe M (1999) The Notion of 'Classes of a Path' in an ER Schema, Proceedings of 3rd East European Conference on Advances in Databases and Information Systems (ADBIS'99), Maribor, Slovenia. Lecture Notes in Computer Science 1691. Springer, Berlin
11. Howe D (2001) Data Analysis for Database Design, 3rd edn, Butterworth-Heinemann, Oxford
12. Hacid MS, Petit JM, Toumani F (2001) Representing and reasoning on database conceptual schema. Knowledge and Information Systems, 3, pp. 52–80
13. Hutchinson SE, Sawyer SC (2000) Computers, Communication and Information: A User's Introduction, 7th edn, McGraw-Hill Higher Education, Boston, MA.
14. Hull R (1986) Relative information capacity of simple relational database schemata. SIAM Journal of Computing, 15(3), pp. 856–886
15. Imielinski T, Spyratos N (1984) On lossless transformation of databases schemes not necessarily satisfying universal instance assumption, Proceedings of the 3rd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems. Waterloo, Ontario, Canada, pp. 258–265
16. Kwan I, Fong J (1999) Schema integration methodology and its verification by use of information capacity. Information Systems, 24(5), pp. 355–376
17. Lim E, Chiang RH (2000) The integration of relationship instances from heterogeneous databases. Decision Support Systems, 29, pp. 153–167
18. Levene M (1998) On the information content of semi-structured databases. Acta Cybernetica, 13, pp. 257–275
19. Lee D, Mani M, Chiu F, Chu WW (2002) NeT and CoT: Translating Relational Schemas to XML Schemas Using Semantic Constraints. CIKM'02. McLean, Virginia, pp. 282–291
20. Mingers J (1995) Information and meaning: foundations for an intersubjective account. Journal of Information Systems, 5, pp. 285–306
21. Miller RJ, Ioannidis YE, Ramakrishnan R (1993) Understanding schemas., Proceedings of Research Issues on Data Engineering: Interoperability in Multidatabase Systems, Vienna, Austria, pp. 170–173
22. Miller RJ, Ioannidis YE, Ramakrishnan R (1993) The use of information capacity in schema integration and translation, Proceedings of the International Conference on Very Large Data Bases. Dublin, Ireland, pp. 120–133

23. Miller RJ, Ioannidis YE, Ramakrishnan R (1994) Schema equivalence in heterogeneous systems: bridging theory and practice. Information Systems, 19(1), pp. 3–31
24. Miller RJ, Ioannidis YE, Ramakrishnan R (1994) Schema intension graphs: a formal model for the study of schema equivalence. Technical Report 1185, Department of Computer Sciences, University of Wisconsin-Madison, WI
25. McBrien P, Poulovassilis A (1998) A formalisation of semantic schema integration. Information Systems, 23(5), pp. 307–334
26. Poulovassilis A, McBrien P (1998) A general framework for schema transformation. Data and Knowledge Engineering, 28, pp. 47–71
27. Stamper RK (1973) Information in Business and Administrative Systems, Batsford, London, pp. 14–22, 324–325
28. Stamper RK (1996) Signs, information, norms and systems. In: Holmqvist B, Andersen PB (eds.), Signs at Work, De Gruyter, Berlin
29. Stamper RK (1982) Organizational semiotics. In: Mingers J, Stowell F (eds.), Information Systems: An Emerging Discipline? McGraw-Hill, London
30. Tsichritzis D, Lochovsky FH (1982) Data Models. Prentice-Hall, Englewood Cliffs, NJ
31. Ullman JD (1982) Principles of database systems, Computer Science Press, Potomac, MD
32. Zhao JL (1997) Schema coordination in federated database management: a comparison with schema integration. Decision Support Systems, 20, pp. 243–257

Chapter 10
Towards a Social-Based Process for Information System Development: A Case Study

C.A. Cocozza Simoni, A. Meincke Melo, M.C.C. Baranauskas

IC – Institute of Computing, Unicamp – State University of Campinas, Caixa Postal 6176, 13084-971, Campinas, SP, Brazil

Abstract

The role played by the computer in organizations continues to evolve and increases in importance, since it mediates social relationships. To improve the information system (IS) development process we need a better understanding of the organizations and their internal and external interactions and dynamics. This chapter aims at discussing a semiotic-based approach to the development of IS. The proposed approach is illustrated with a case study in which a real organization was exposed to methods of organizational semiotics (OS) to rethink its way of developing systems. This work has allowed us to verify the contributions of OS to the redesign of an IS development process in an IT organization.

10.1 Introduction

Rethinking an organization is one way to improve business performance, understanding what it is doing now, identifying the essential activities to be performed, the stable organizational behaviour, and the changeable activities, normally related to message and control activities [8]. This process allows the organization to focus on the changeable part, to find how to perform it in order to improve effectiveness.

The result of this approach can change the organization or its business process in several ways, including simplification and improvement of its

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 177–192.

processes, redesign or re-engineering of the business process, redesigning of the business scope, or a corporate transformation [1].

In a contemporary society the role played by software has increased in importance, but it remains a difficult issue "simply summarized as software taking too long to develop, costing too much, and not working very well when eventually delivered" [5, p. 65]. The same authors suggest that "a disciplined approach to software development through the use of software development methods could help address these problems", and despite the existence of "hundreds of such commercial or brand-named software development methods, these are not widely used in practice, and are certainly not used in their entirety". They consider that the use of methods should be flexible and tailored to the actual needs of the development context.

The role played by the computer is very important, because it mediates social relationships [1]. "When re-engineering we must understand that we are re-engineering a social system and not a deterministic mechanical process" [1, p. 4]. Moreover, to improve effectiveness in an organization it is necessary to involve not only questions related to costs, quality, and services, but also related to effective cooperation of the organization's resources and its partners (e.g. suppliers, clients, and government). These are not questions to be easily addressed in traditional development processes [10].

Many authors have shown the significant role of social and organizational aspects in our interactions with technology [3, 6, 9, 10, and 16], and the influence of these aspects must be taken into account in the analysis and the design of systems. Literature in organizational semiotics (OS) has shown that the social, cultural, and organizational aspects involved in the problem must have a more decisive role in the process of developing IS, while traditional methods have emphasized the technological solution itself. The main assumption behind the first approaches to the development of the technical information system (IS and the traditional methods from software engineering (SE) can be characterized by a strong belief in systematic design methods based on mathematical and logical theories, suggesting that the users (end-user, client, customer, stakeholder, or problem owner) are supposed to give complete and explicit descriptions of their demands in terms of the system to be developed [4, 7, 10, 12, 15].

The better of the two worlds seems to be necessary to a broader understanding of the problem of developing IS that make sense to their users in their organizational contexts. "Organizational change belongs to the social science …, and we cannot simply use the methods in the natural science to observe and judge the results of social science" [10, p. 5]. Our previous work investigated the use of the OS methods in a combined way with the unified process (UP), to compose a complete cycle of IS development. We have been practising OS and UP techniques together [2, 14], as well as OS within a traditional system development cycle [13]. The first outcomes have shown that this practice has allowed the analysts, together with the problem owners and stakeholders, to have a deeper understanding of the problem and its context, leading to potentially more meaningful solutions.

This chapter presents and discusses the method and activities carried out for building a social-based IS development process for the IT Department of the General Administration of our University (DGA-AT), considering the social, political, cultural, and ethical issues involved in the understanding of the IS development context of this department. Section 2 presents some key concepts of the OS methods that have a role in this work. Section 3 presents and discusses the case study and the proposed approach, and section 4 concludes the chapter.

10.2 Background

One of the arguments for OS is based on the hypothesis that all organized behaviour is affected by the communication and interpretation of signs by people. OS understands the internal activities of an organization, including its IS and its interactions with the environment, as a semiotic system [9]. The case study reported in this chapter is based on problem articulation method (PAM), one of the methods for eliciting, analysing, and specifying user requirements (MEASUR) methods, to be applied in the initial phase of a project, when the problem definition is still vague and complex.

In OS, an organization can be seen as an IS in which interdependent links between the organization, the business process and the IT system occur [9]. At an informal level there is a subculture where meanings are established, intentions are understood, beliefs are formed, and commitments with responsibilities are made, altered and discharged. At a formal level, form and rule replace meaning and intention. At a technical level, part of the formal system can be automated by a computer-based system. The different levels of the IS are organized in such a way the informal level embodies the formal, which by its turn embodies the technical level. The IS has impact in and reacts to the environment. In a semiotic perspective, different layers of meaning must be considered in the IS analysis and software design. PAM is used to understand the forces involved (needs, intentions, existing conflicts, etc.) among the stakeholders, allowing a big picture of the problem context and the main requirements.

Figure 1 shows the organizational onion that represents the internal and external relationships of an organization.

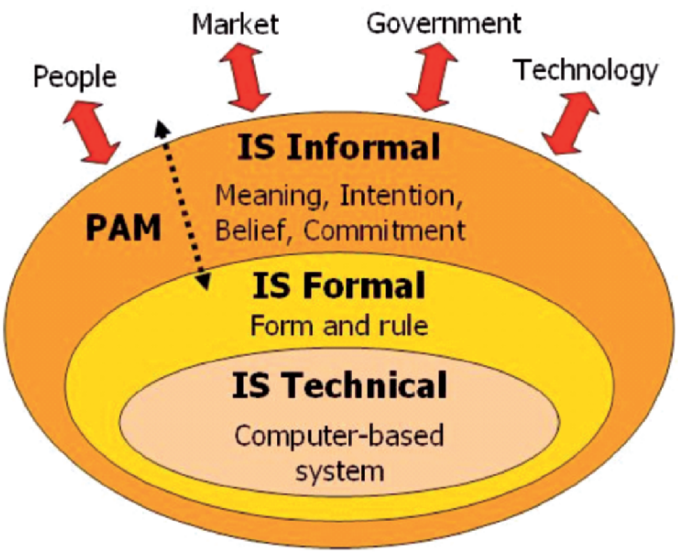

Fig. 1. The semiotic onion situating PAM collaboration

Our hypothesis is that the OS methods, combined with concepts, methods, and techniques from SE, human-computer interaction (HCI) and participatory design (PD), allow social and technical understanding of IS development process, once these areas together can provide a framework for building a formal process which reflects different points of view (e.g. technicians, clients, and/or users).

While OS and HCI have provided theoretical background, methods, techniques, and guidelines to the design, evaluation and implementation of interactive computing systems for human use, PD approach, in particular, has provided theoretical background and techniques to the design process of products with the direct participation of their users. In a PD perspective, the product is not only designed for the users, but in collaboration with them. Researches in PD have shown different ways to include end-users in the process of designing technology [11]. At the same time the user participation is considered valuable to reach product quality, as it allows a better understanding of activities and context of work by the combination of different experiences. The PD approach can also be useful to the users themselves, inspiring them to think about and analyze their own process of work.

In this work a case study related to the interests of DGA-AT is presented and discussed.

10.3 From Software Development to IS Development: A Case Study

The General Administration of our University (DGA) is a department responsible for supplies (buying and stocking), import, finance, accounting, and general services at Unicamp. It has an internal IT area (DGA-AT), which is responsible for software development, IT infrastructure implementation and support. DGA-AT deals with a very heterogeneous group of internal clients, with specific needs, language, interests etc.

Nowadays DGA-AT is reviewing the existing DGA web-based Portal, which allows the internal and external community to access the services provided by the department. Starting from a very informal work process, focused on technical issues and dependent on individual knowledge and initiatives, DGA-AT was interested in a complete evaluation of their way of working with software development. Their goal was to establish a process adapted to their needs, culture, and characteristics, aspects not very well known even for people working there. Thus, we proposed the use of PAM to evaluate their current methods and procedures, aiming at achieving a formal process as a result of our analysis.

10.3.1 The informal IS – understanding the context of an IS development

PAM was carried out in the format of two-hour workshops, with the participation of co-workers identified during a meeting where a commitment about how to conduct this process was made. The participants were the manager of DGA, the manager of DGA-AT, one user representative, one web designer and two IT technicians. Table 1 shows the initial agenda for the meetings, the employed PAM techniques, and the involved participants.

The discussions were reported in formal documents, and the notes and the outcomes of each method were documented. In the workshops we used OS artefacts (e.g. Stakeholders Frame, Evaluation Frame, Workflow Chart, Functional Morphology Frame, and Semiotic Frame) in a collaborative and participatory format. The main outcome of the workshops was the agreement on requirements including considerations on the DGA-AT goals and commitments, management and quality of processes, the need for better communication with DGA areas (including end-users), and technical issues of the IS development process.

Table 1. PAM into practice

Meeting	PAM technique	Stakeholders
0	The first meeting	The manager of DGA, one user representative, one web designer and one IT technician.
1	Stakeholder analysis	One user representative, one web designer and one IT technician.
2,3	Evaluation framing	One user representative, one web designer and one IT technician.
4,5,6,7	Morphologic analysis	One web designer, two IT technicians, the manager of DGA.
8	Semiotic diagnosis	One IT technician and the manager of DGA.

In between the workshops, the DGA-AT manager, technicians, and the user representative reviewed the outcomes of each meeting by themselves and prepared the material for the next meeting. This procedure allowed us to observe how much of the techniques were understood, the way they made use of them, etc.

Some results from the use of PAM techniques are briefly described in the following sections.

Stakeholder Analysis

This analysis allows us to investigate the interested parts (the stakeholders) that directly or indirectly have influences or interest in the IS under consideration. The Stakeholder Frame produced as a result of the first meeting is shown in Fig. 2. DGA-AT people realized that there were more interested parts with different roles in the IS development than they used to consider, such as other areas within the university and people outside the university who could have interest in information from DGA.

Evaluation Frame

It allows us to identify, for each stakeholder, his/her interests, questions, and problems, in order to discuss new ideas and possible solutions. They discussed and recorded the expectancies, problems, and questions involved in the development process related to each stakeholder. It was important that the discussion not only related to technical issues, but to the need of a better knowledge of other DGA areas, their need to be more purposeful, inventive, innovative, and so on. Table 2, in Appendix, shows some outcomes of this analysis.

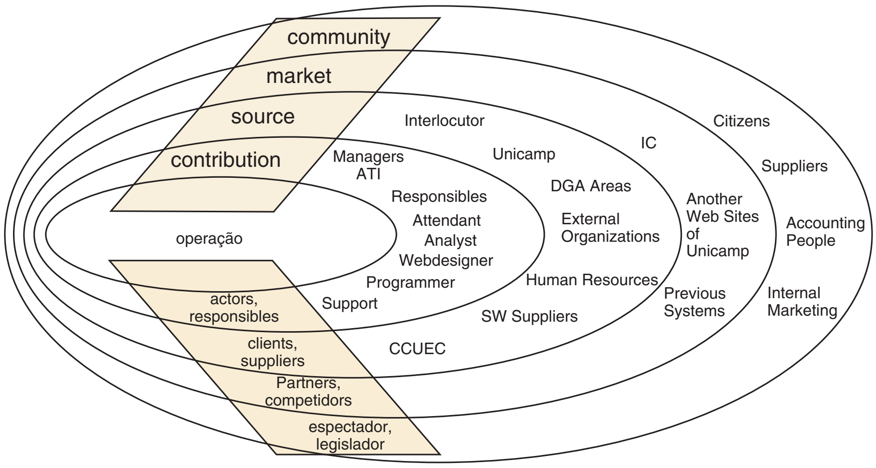

Fig. 2. The Stakeholder Frame produced

Morphologic Analysis

It allows the investigation of the morphology of the organization tasks and functions. Three main components of the analysis are the substantive, which focus on aspects that contribute directly to the organizational objectives; the communication, which is used to inform people and coordinate actions; and the control, which is used to reinforce the whole business system running properly.

By using this technique combined with Artifact Walkthrough from PD [11], DGA-AT people realized, for the first time, they had an ad hoc process, and they could explore its weaknesses, decision-making behaviour, etc. We perceived that the communication and control components were extremely informal and weak. This analysis took us four sessions allowing the capture of views of the different parts involved (e.g. managers, technicians and users). Table 3, in Appendix, shows some outcomes of this analysis.

Semiotic Diagnosis

Traditional system development methodologies emphasize technical issues (physical world, empirics, and syntactic aspects) and the analyst misses the opportunity of understanding other levels of relationship given by the semantic, pragmatic, and social levels of the Semiotic Framework, which directly or indirectly affect the system design. The use of the Semiotic Framework allowed us to examine the organization as a social system that is established through the use of information.

By using this technique the participants could align the organizational goals and commitments, as well as to identify needs, problems, and possible solutions from social to technical aspects. They identified the need to know the other DGA areas, their meanings to all the involved things; the need to update data and resources. Table 4, in Appendix, shows some outcomes using this technique.

As a feedback on this phase they considered their experience with PAM very positive and were "surprised with the results"; they became interested in applying these techniques in their future projects. Regarding the new process, they considered it as an opportunity for the other DGA areas to realize that the software development should not be done without formalism, planning, and commitment among all involved parts. The Semiotic Framework had specially caused a good impression, mainly to the DGA-AT manager, by its possibility of synthesizing coherently in a unique document different aspects of IS development process.

10.3.2 The formal and technical IS – structuring an IS development process

Based on these preliminary results of using PAM and on previous works [2, 13–14] a cyclic IS development process emerged, in which there is no separation between the development and the maintenance stages. The idea behind it is to provide a continuous improvement of their development process and products, by evaluating the process at the end of each iteration, and by a systematic evaluation on the product, shared between DGA-AT and end-users.

The conceptual model of the process, shown in Fig. 3, is grounded on several disciplines: OS, SE, HCI, and PD, expressed in the bottom layer. Methods, techniques, and recommendations, expressed in the middle layer of Fig. 3, were investigated to structure the cyclic process model for software development, showed in the upper layer of Fig. 3. It was not a simple formalization of the current work process nor an implementation of a standard development process, but a new and detailed formal work process was modelled according to the main goals and commitments elicited with the stakeholders of this case study, keeping some cultural and social aspects of the current relationship between DGA-AT and their clients.

While OS, through MEASUR, offers a basis to the earlier steps of the process (e.g. Context Analysis, Planning, Negotiation and Requirements Elicitations), SE offers methods and models that could be adopted for completing software development process from a technological perspective (e.g. UP, UML, V & V, quality, project management). Moreover, HCI gives us guidelines and techniques for interface design and its evaluation.

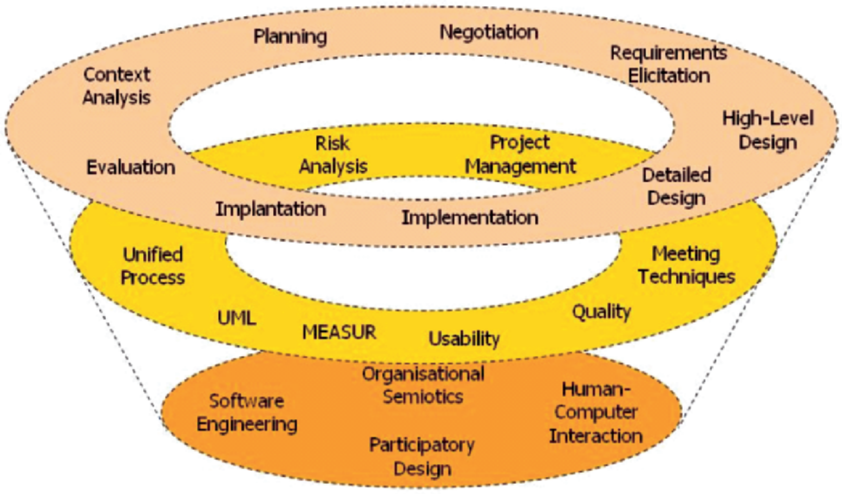

Fig. 3. The technical, methodological, and theoretical basis of the approach

Having OS and HCI as background, end-user involvement is considered essential. Thus, based on their interests, PD has allowed us to suggest techniques to involve users in the whole process, from the early stages to the first deployment of the technical system. Complementing the responsibility of each stakeholder was defined in each activity of the process.

The process of choosing techniques, establishing recommendations, and formalizing and documenting the development process is now in progress with the design of the DGA portal.

10.3.3 Discussion

This case study allowed us to verify the contributions of OS in redesigning the work process of this group. By using PAM we could capture information regarding the previous DGA-AT process, including social, behavioural, ethical, and political aspects, which would not usually be captured by traditional methodologies. A detailed list of agreed requirements for the proposed process was derived from the artefacts used in the meetings, and allowed a deeper consideration of the semantic, pragmatic, and social levels, as well an understanding of the social practices to be supported.

In this first phase, people related to the software development process pondered deeply on their work practices, presented problems, and their wishes, and proposed new ideas and solutions for the identified problems, considering all stakeholders involved. It was remarkable their desire for a better systematization and formalization of their activities, as well as the need for a deeper and wider user involvement and commitment. Their current way of working seemed not to allow this kind of reflection.

We could identify some desirable features of the new process not only related to technical issues, such as: processes documentation, environment reorganization, more formal way of communicating with users, meetings during the project with project managers, technicians and representative users; a better interface between analysts and web designer; a better project specification; a way to detect system failures, usability recommendations, user feedbacks, system problem history, and so on. The semiotic framework allowed us to organize these ideas very well, distributing them in the six layers, aligned to the main goals and commitments elicited in the social world layer.

Figure 4 shows, in a synthesized way, the old workflow and the new one achieved after improvement in the DGA-AT product development process.

Label A, in Fig. 4, represents the planning activities, which in the old process is carried out independently of the problem complexity, after a shallow problem understanding. It used to deal with the resources involved in the problem and it used to structure a first draft of schedule for problem solution. Using PAM they realized the necessity of a more formal problem identification and clarification. Even considering that PAM has been recommended for more complex projects, the group decided to use it to cover these first activities. It was agreed that at least a checklist based on PAM should be used in small projects. Another aspect covered in the new process is the necessity of commitment between DGA-AT and the users of its products, requiring a formal proposal to be accepted or rejected by the users. The role played by the users is changing in the new process, and the establishment of steering and key user committees, which will be the communication channels between the areas and DGA-AT, is in progress.

Label B shows they are moving from a scenario where they "scribbled" projects to a new one using PAM, Semantic Analysis (SAM), Norm Analysis (NAM), and some techniques from SE, to construct a high-level problem specification with more formalism and richness of details.

Label C represents a more formal project specification as a process for testing and validating the solution in the new scenario. The concept of iterative and incremental design to deliver the project in parts was also introduced.

Finally, Label D, introduced in the new process, shows activities related to quality, to evaluate both the development process and the product delivered, allowing to close the development cycle, and to start a review of the process as well as of the product.

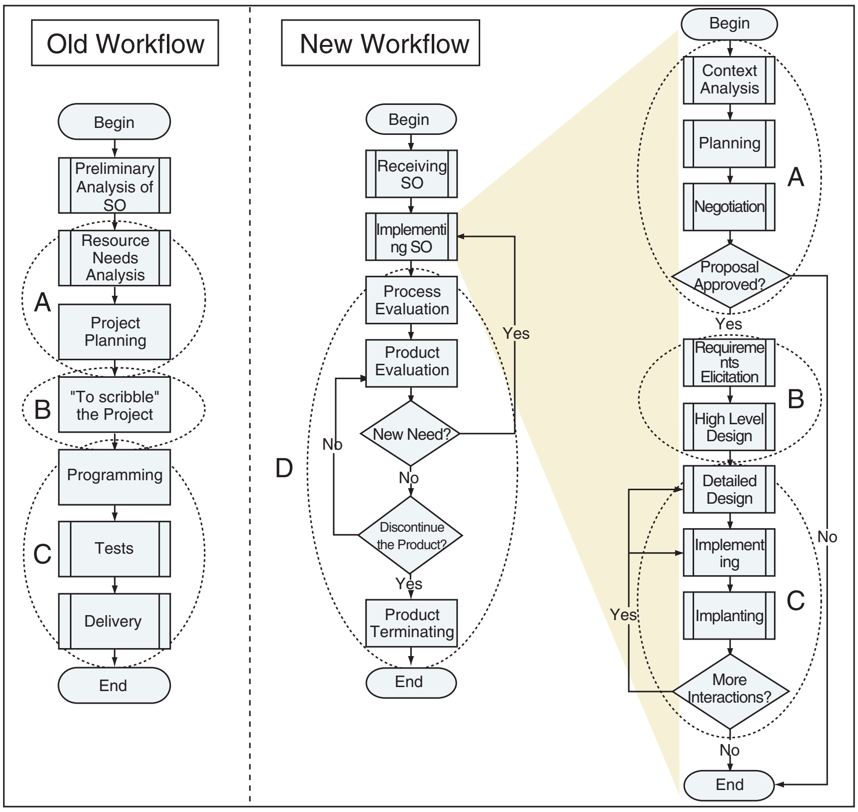

Fig. 4. The old and the new workflows face-to-face

An important result from this case study was the use of OS methods both for analyzing the DGA-AT work process and as a part of the IS development process. This allowed the use of a same language inside DGA-AT, to review and to maintain the development process, as well as to be used with clients and/or users during the analysis of their IS and the development of computer-based solutions. Furthermore, the approach allowed exposition and exploration of an OS-based approach in a real context. The characteristic of each participant in the group allowed us an exercise of empirical verification of the effectiveness and acceptance of the approach by technicians and users.

Currently DGA-AT is experiencing the use of PAM, in a web-based portal, and they have already realized the need to contact users and areas of DGA to elucidate some unsolved questions from a first interaction they had through questionnaires, using the previous process, before rethinking their way of work. Some new questions are: What is wrong with the existing portal? What are the expectancies of the areas related to the portal? Who is the public for each area? What is the relevant information to the users of each area? Is there some website that could be referenced to the design of a new portal? What should be a good policy for updating the content of the portal? What does each area think about customization? What kind of knowledge is required from the users when they are dealing with the information published in the portal and how to provide such knowledge?

10.4 Conclusion

This work dealt with the process of rethinking an organization, in our case the IT department from the general administration of our university, mainly considering the social and technical aspects involved in it. We could also investigate the role played by OS in providing instruments to link these different aspects in the same process of system development.

The case study allowed us to verify the contributions of OS to the redesign of an IS development process in an IT organization. With OS, SE, HCI and PD approaches taken together we could structure a process “tailorable” to a specific domain. Techniques from each of these disciplines could support DGA-AT in developing and maintaining software, experiencing the potentiality of OS in constituting bridges between them.

The final results encourage further work towards a formalization and utilization of this approach, verifying and establishing relationships between OS and other areas as SE, HCI and PD. Finally, practical work in DGA continues; they are now detailing the technical layer while validating the proposed process in the development of a web-based portal for DGA. This step will allow us to verify the influence of the approach in the quality not only of the software application, but of the business process as well.

References

1. Blyth A (1997) Business Process Re-engineering: What is it? In SIGGROUP Bulletin – April, 18(1)
2. Bonacin R, Baranauskas MCC, Liu K (2004) From Ontology Charts to Class Diagrams: Semantic Analysis Aiding Systems Design, The 6th International Conference on Enterprise Information Systems, Setúbal, Portugal
3. Dourish P (1995) Developing a reflective model of collaborative systems. ACM Transactions on HCI, 2(1), March 1995

4. Ehn P, Lowgren J (1997) Design for quality-in-use: human-computer interaction meets information systems development. In: Helander M, Landauer TK, Prabhu P (eds.), Handbook of Human-Computer Interaction, 2nd edn, Elsevier Science, Amsterdam
5. Fitzgerald B, Russo NL, O'Kane T (2003) Software development method tailoring at Motorola. Communications of the ACM, April, 46(4)
6. Heusden B, Jorna RJ (2000) Reconsidering the Standard: A semiotic Model of Organization(s), Groningen University, Groningen, New Zealand
7. Jacobson I, Booch GB, Rumbaugh J (1999) The Unified Software Development Process, Addison-Wesley, Reading, MA
8. Liu K, Stamper RK, Huang K (1997) A morphology for re-engineering competitive business. In: Booth A (ed.), Legacy to Client/Server – Have You Chosen Wisely, UNICOM Press, New England, USA
9. Liu K (2000) Semiotics in Information Systems Engineering, Cambridge University Press, Cambridge
10. Liu X (2001) Employing MEASUR Methods for Business Process Reengineering in China, Ph.D. thesis, University of Twente, Enschede, The Netherlands
11. Muller MJ, Haslwanter JH, Dayton T (1997) Participatory practices in the software lifecycle. In: Handbook of Human-Computer Interaction, 2nd edn, Elsevier Science, Amsterdam, pp. 255–297
12. Pressman RS (2001) Software Engineering: A Practitioner's Approach, 5th edn, McGraw-Hill, New York
13. Simoni CAC, Baranauskas MCC (2004) The Practice of Software Development and the Organizational Semiotics Approach: A Case Study in Business Organizations, The 7th International Workshop on Organizational Semiotics, Setúbal, Portugal
14. Simoni CAC, Bonacin R, Baranauskas MCC (2005) Bringing Social Constructs to the Information System Development Process: Contributions of Organizational Semiotics. ICEIS/2005 – The 7th International Conference on Enterprise Information Systems, Miami, USA
15. Sommerville I (2001) Software Engineering. In: International Computer Science Series, 6th edn, Addison-Wesley, Boston, MA
16. Stamper R, Liu K, Hafkamp M, Ades Y (2000) Understanding the roles of signs and norms in organizations – a semiotic approach to information system design. In: Behaviour and Information Technology, 19(1), pp. 15–27

Appendix: Some Outcomes from the Use of PAM Techniques

In the next pages we show some outcomes from the use of PAM in the process performed in DGA-AT.

Table 2. Some evaluation frame outcomes

Stakeholder Frame Level	Stakeholder	Expectancies	Questions/problems
Contribution	DGA Manager	DGA-AT personnel should be more purposeful	DGA-AT people are more reactive today
	DGA-AT Manager	Improvement of periodic meetings	Some in progress activities are missing management
	Analyst	Improvement of users involvement	Absence of a formal methodology with the role of each participant
	Web Designer	Detailed description from analysis	No customization is offered in the website
Source	DGA Areas	Facilities to find and to maintain information	Data updating without audit trail
	Unicamp	Facilities to find and to maintain information	Non updated information
Market	Another web sites of Unicamp		Absence of patterns and documentation
	Previous systems		
Community	Citizen	Facilities to find information	Difficult to access the interested information

Table 3. Some morphologic analysis outcomes, in a macro process flow

Process of development and maintenance system				
Substantive			Message in general	Control in general
Substantive	Message	Control		
To make a service order (SO)	To give return to the users	To determine whether to execute the SO	Interact with the groups according to necessity	To follow the SO
To analyze the SO	To talk about the SO	To close the SO	To receive and advise about the University informs and resolutions	To review priorities
To obtain more information	To inform users about trainings	To define whether the SO information is enough		To make an annual report
Preliminary analysis		To define the complexity of the SO		
To perform new development		To define resource for the SO		
To perform maintenance		To talk to verify delays		
To identify new needs				

Table 4. Semiotic framework – semiotic diagnosis feedback

Human	Social	Quick attending, to reach user expectation, better user involvement, quality in use, integration between DGA areas, to follow legislation and norms, and updated content
	Pragmatics	Being more purposeful, better attending formalism, systems that allow higher productivity of areas, gathering ideas and user expectation, communication between development team and users, less bureaucracy, agile and easy programs to find information, and every area having access to systems of their interest
	Semantic	Knowing the activities of other areas, knowing the basic activities of DGA-AT, well-defined system objectives and functionality, to find information easily, to access information easily, and manager having an holistic vision about developing systems and their resources
TECHNICAL	Syntactic	Documentation process, environment organization, process of attending, failure caption systematization, suggestions for usability and functionality, adequacy on organizing information, development process, management process, and external software acquisition/installation process
	Empirical	Time to answer, periods of dedication without interruptions, updated information, and updated information about human-resources
	Physical	Channels of communication between development team and users, people, hardware, time, physical, and environmental infrastructure

Part 5
Application of Organizational Semiotics

Chapter 11
A Semiotic Framework for Research into Self-Configuring Computer Networks

J.H. Connolly, I.W. Phillips, L. Hawizy, J.I. Rendo-Fernández

Department of Computer Science, Loughborough University, Loughborough, LE11 3TU, *UK*

Abstract

Self-configuring computer networks are designed to offer services to users in response to their specific requirements on particular occasions. In order for such networks to obtain information about their users' requirements and then to respond appropriately, processes of communication need to take place, not only between the user and the network, but also within the network itself for the purpose both of configuring the network appropriately and of providing the required services. These processes can be analysed in terms of a multilevel semiotic framework in such a way as to clarify our understanding of their properties in relation to structure, meaning, and contextually situated use. Such a framework offers an attractive basis for research into self-configuring networks.

11.1 Introduction

11.1.1 Background

Computer networks have become a very familiar part of the contemporary world. At one end of the scale, local networks play a vital role in supporting the activities of countless organisations, and they have also spread into less formal contexts, such as domestic settings. On a broader front, the

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 195–209.

global Internet has enabled people all over the world to enjoy network access on a grand scale.

The authors of the present chapter are involved in research into networked systems. In particular, we are interested in environments known as *self-configuring* or *ad hoc* networks. In these environments a number of electronic devices are co-located and connected to a common networking medium, which may be either wired or wireless. Within such a created network the devices communicate to determine the services that each can offer; and then the collaboration offers a set of combined services to the user or users. Examples include: (i) a domestic environment, where each device in the home is connected to every other device, allowing for interactions currently not possible; (ii) an operating theatre environment, where devices to support a surgical operation are be brought together to assist the hospital workers; and (iii) a spacecraft environment, where devices that have been manufactured to heavily defined constraints employ a self-configuring network in order to provide services to the users and to the craft itself.

In considering such environments the difference between the devices and the network should be recognised. All the intelligence resides in the former, while the latter is simply a medium for interconnection.

When a device is brought into the network, whether through a physical connection (being plugged in) or coming into the range of a wireless connection, it must communicate its entrance to the other devices. In addition, devices require the ability to enquire about the services that other devices offer and to respond to such enquiries themselves.

Devices therefore provide services to other devices; and the combinations of these services (superservices) provide benefits to the user or users of the network. Many systems exist to provide such interconnection. These include UpnP, HAVI and JINI, while more academic research-oriented examples are Aura (CMU), Oxygen, Endeavour and Portolano. However, accounts of these have not so far contained any explicit discussion of the semiotic implications of such systems.

UPnP (http://www.upnp.org) is based on XML and Internet technologies and describes devices in terms of a set of services with invokable operations. HAVI (http://www.havi.org) is designed for interconnection of home audio and video equipment based on Firewire (IEEE 1394) as a physical layer. It defines services as a set of objects, again with operations. JINI (http://www.jini.org) is a Java-based service-discovery mechanism. Services are specified as Java interfaces and the interaction between clients and services is done through remote method invocation (RMI).

In Aura [12] the system exists to support user mobility; for example as a user moves between environments it allows the user to continue on a task. The task is the main focus in this system. Oxygen (http://oxygen.lcs.mit.edu) is a mix of technologies to provide a pervasive computing environment. Data in the system are represented as objects, with other objects represented through named arcs. Endeavour's (http://endeavour.cs.berkeley.edu) main concepts are those of mobile software and nomadic data, the system being responsible for the management of moving code and data. Portalano [9] is a system focusing on the satisfaction of user needs.

11.1.2 Aims of the present chapter

The research interests of the present authors extend not only to networks but also to organisational semiotics (OS). Thus, it seems natural that we should explore the possibility that OS might provide a theoretical framework in terms of which to conduct our work on self-configuring networks. This would hold the promise of a novel approach to the subject.

Accordingly, the purpose of the present chapter is to address the following questions:

1. Does OS lend itself as an appropriate and useful theoretical framework within which to conduct research into self-configuring networks?
2. If so, then what advantages may it offer?

Let us begin by considering the first of these questions.

11.2 The Framework and its Application

11.2.1 Communication processes

Semiotics comes into its own when it enables us to understand processes of *communication*. Hence, the first issue that needs to be clarified is the following:

3. As far as self-configuring networks are concerned, what process or processes of communication are involved?

In fact, there are at least three such processes:

4. Communication in relation to networked systems:
   - (a) The interaction that is conducted between user and system.
   - (b) The interaction that takes place among the devices within the network itself, in order to accomplish tasks which the system serves to facilitate and support for the benefit of the users.

(c) The communication that is involved in the work of developing and maintaining the networked system. This work includes configuring and reconfiguring the network.

Ultimately, we need to achieve a thorough understanding of all three processes. However, in the present chapter we shall be concerned in particular with (4a) and (4c). As in our previous work in OS [3–7, 11], we shall make use of the six-level framework proposed by Stamper [13].

11.2.2 Interaction between user and networked system

It transpires that the analysis of (4a) in terms of the six-level framework delivers a quite different result from that of (4c). The analysis of the interaction between user and system reveals the following:

5. Analysis in terms of semiotic levels:

(a) Physical world:
The physical support or basis for the communication is supplied by the hardware of the networked system. In the case of spoken interaction, the air through which the sound travels also has a role to play.

(b) Empirics:
The physically observable activity found in the electrical signals that flow across the network. In the case of spoken interaction, air-pressure waves in the atmosphere also play their part.

(c) Syntactics:
The communication may be based on various possible semiotic systems, for instance a formal language (such as the command language associated with the Linux operating system) or a natural language. The (abstract) structure of the communicated message is determined by the grammar of the semiotic system concerned.

(d) Semantics:
Similarly, the meaning of the communicated message is determined by the semantics of the semiotic system employed.

(e) Pragmatics:
At this level we consider the following processes:

(i) The exchange of intelligible information between user and system.

(ii) The resultant achievement (hopefully) of the user's intentions that lie behind and motivate the communicative activity.

(f) Social world:
The social context of the interaction could be, for instance, the carrying out of an operation within a hospital or the sharing of data within a research establishment.

Let us now compare this with the analysis of the communication involved in configuring and reconfiguring networks.

11.2.3 Communication relating to network (re)configuration

In order to bring out the contrast between the two analyses as clearly as possible, let us first consider the situation where a human engineer is going to (re)configure a network manually with the help of a diagram, which may be drawn on paper or displayed on the screen of a computer which need not be connected to the network in question. The diagram constitutes a representation that serves to encapsulate the design or plan of the network. Hence, it is a semiotic object, and when analysed in terms of the six-level framework it may be viewed in the following manner:

6. Analysis in terms of semiotic levels:

(a) Physical world:
The paper or screen on which the diagram is manifested.

(b) Empirics:
The observable, perceptible manifestation of the diagram itself.

(c) Syntactics:
Under this heading there are three semiotic considerations:

(i) The basic elements of the diagram, namely the nodes and arcs of the network diagram.

(ii) Any labels serving as annotations to the network diagram. These may well be in the form of natural language.

(iii) Any constraints imposed upon the combination of elements.

(d) Semantics:
The meaning of the elements in the representation. Nodes denote devices while arcs denote connections.

(e) Pragmatics:
The communication of the network configuration from the originator of the diagram to the reader (which could be the same person at a later time, in which case we speak of self-directed or *reflexive* communication).

(f) Social world:
The accomplishment of tasks that human beings such as engineers (considered as members of social communities and organisations) want done, for instance the (re)configuration of a network inside a spacecraft.

In order for a network-based system to be able to (re)configure *itself* automatically, it will need to be provided with an internal representation of the network concerned. (For the sake of simplicity, we shall suppose that this representation is stored on just one of the networked devices. Distributed or mirrored representations are not the concern of the present chapter.) The representation will correspond conceptually to the engineer's diagram, but will be in a machine-tractable form. Its semiotic analysis will therefore be along the following lines:

7. Analysis in terms of semiotic levels:

(a) Physical world:
The hardware in which the representation is stored.

(b) Empirics:
The internal state of the storage device. (This presupposes no visual display of the network configuration; however, see below).

(c) Syntactics:
As for (6c).

(d) Semantics:
As for (6d).

(e) Pragmatics:
The use of the representation in automatic reasoning activity, whereby the network-based system works out how to (re)configure itself.

(f) Social world:
As for (6f), except that the engineer may not now be part of the scene.

The representation will be used during the automatic reasoning process, which can be regarded as a kind of reflexive communication.

It is also possible that the system will be equipped with a facility for displaying the current network configuration to the user. If so, then the display is likely to be in the form of either a diagram with the semiotic characteristics identified in (6) or else some kind of textual paraphrase. From the point of view of readability, the diagram is likely to be preferable.

11.3 Multilevel Analysis and Description

When a communication process is subjected to semiotic analysis, the result is a stratified description that enables us to consider one aspect of the process at a time (insofar as this is useful; it is acknowledged that there are interdependencies between the different levels). What is more, it opens the door to the application of existing work in semiotics to the phenomenon that has undergone the analysis, when the applicability of such work might otherwise have escaped notice.

OS has already been applied to the related area of pervasive computing by Andersen [1] and Brynksov and Andersen [2]. This has shown the promise of OS as a useful framework for modelling in this field.

In the context of the present chapter, a specific benefit of applying semiotic analysis to self-configuring networks is that it leads to a view of these networks as syntactical structures with semantic interpretability and pragmatic potential. Let us now try to develop this idea.

On a matter of terminology, we shall henceforth employ the term "plex" to denote a network considered from a specifically syntactical point of view. The term "plex" has been borrowed from database theory, but its use in the present context is unrelated to databases.

11.3.1 Syntactics

Regarding a plex as a syntactical structure encourages us to think of it in terms of a grammar. This idea has a precursor in the syntactic approach to pattern recognition [10], in which visual objects are described by means of grammars whose primitives are shapes such as vertical or horizontal lines. A grammar is basically a definition of how elements may be combined to form well-formed structures, satisfying a set of constraints. The parts of which a given structure consists are called its constituents. If a structure is analysed into smaller and smaller constituents until no further analysis is possible, then the analysis will have reached the ultimate constituents of that structure.

In the case of a network diagram, the ultimate constituents, or primitives, are (i) the nodes of the network and (ii) the arcs (representing connections) by which they are linked. The simplest structure then consists of two nodes and the arc that joins them. We shall term such a structure a "linkage". If we assume that a given node may be a constituent of more than one linkage, then we can regard the entire plex as being composed of linkages. This suggests a grammar along the following lines:

8. Plex -> Linkage^+
Linkage -> Node + Arc + Node

The arrow symbol "->" may be read "can consist of". Hence, the first of these two rules states that a plex can consist of one or more linkages, and the second that each linkage can consist of a node, an arc, and another node. Admittedly this is a very simple grammar, but it will serve the purpose of illustration. The details of the grammatical representation remain a matter for research. However, (8) will suffice to offer a basic idea of the approach. Indeed, it should be noted that although (8) contains only two rules, it nevertheless generates an infinite number of possible plexes.

Building upon common practice in natural-language grammar, we can, if we wish, enhance the elements in our grammatical rules by adding features to some or all of them, along the following lines (the notation being highly provisional):

9. $\text{Linkage}[\text{id} = l_h]$ -> $\text{Node}[\text{id} = n_i] + \text{Arc}[\text{id} = a_j] + \text{Node}[\text{id} = n_k] \mid n_i \neq n_k$

Here we have added a feature to each element, enabling it to be assigned its own identity (id). For instance, the identity of the linkage is denoted as l_h, where h is an integer variable; in the description of an actual network, the linkages will be given identities such as l_1, l_2 and so on. The rule is accompanied by the constraint "$\mid n_i \neq n_k$", which means "given that nodes n_i and n_k have different identities".

Given the above apparatus, we now find ourselves with two significant capabilities. Firstly, we can give a general definition of a well-formed plex by means of a feature-based grammar:

10. Plex -> $\text{Linkage}[\text{id} = l_h]^+$
$\text{Linkage}[\text{id} = l_h]$ -> $\text{Node}[\text{id} = n_i] + \text{Arc}[\text{id} = a_j] + \text{Node}[\text{id} = n_k] \mid n_i \neq n_k$

Secondly, we can describe a particular, individual instance of a network configuration in terms of the structures generated by the grammar. For instance, consider a small star network as shown in Fig. 1.

This network plex contains four nodes:

11. The nodes in Fig. 1:
 (a) n_0 (the hub)
 (b) n_1, n_2, and n_3

These take part in three linkages:

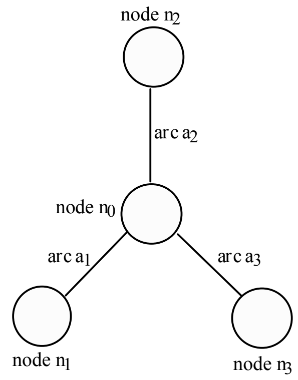

Fig. 1. A simple star network

12. Linkages in Fig. 1:
    (a) Node[id = n_0] + Arc[id = a_1] + Node[id = n_1]
    (b) Node[id = n_0] + Arc[id = a_2] + Node[id = n_2]
    (c) Node[id = n_0] + Arc[id = a_3] + Node[id = n_3]

The syntactical description of the whole plex is therefore as follows:

13. (Plex
    (Linkage[id = l_1]
        (Node[id = n_0]+Arc[id = a_1]+Node[id = n_1]))
    (Linkage[id = l_2]
        (Node[id = n_0]+Arc[id = a_2]+Node[id = n_2]))
    (Linkage[id = l_3]
        (Node[id = n_0]+Arc[id = a_3]+Node[id = n_3])))

With regard to automatic processing, it would appear that, given a grammar such as (10) and a list of linkages such as (12), it would be possible for a system (i) to check whether a network is a well-formed plex, and also whether it remains well-formed following reconfiguration or damage, and (ii) to compute a description of the syntactical structure of the network plex as in (13). There is, of course, an evident analogy with natural-language parsing here, though it is not yet clear to us what is the best choice of algorithm for carrying out the tasks involved.

11.3.2 Semantics

In order to help bring out the difference between the syntactics and the semantics, we have based our structural description of networks on terms like "plex", "node" and "arc", which give no indication of the fact that they are intended to represent computer networks, as opposed to (for example) cities and roads on a map. However, when we move to the semantic level we do need to choose appropriate, meaningful appellations. Moreover, we need to link these interpretations explicitly with their structural counterparts. This can be accomplished by means of a set of semantic interpretation rules (again on the analogy with natural-language processing), as follows:

14. Semantic interpretation rules:

(a)	Plex	->>	Network
(b)	Linkage	->>	Connection
(c)	Node	->>	Device
(d)	Arc	->>	Via

The arrow symbol "->>" may be read "is to be interpreted as". Hence, rule (14a) means "a plex is to be interpreted as a network", and so forth.

It would be possible to apply these rules automatically to convert structural descriptions into corresponding semantic descriptions. For instance, the first linkage in (13), namely (15a), would be semantically interpreted as (15b):

15. Application of semantic interpretation rules:

(a) (Linkage[id = l_1]
(Node[id = n_0] + Arc[id = a_1] + Node[id = n_1]))

(b) (Connection[id = l_1]
(Via[id = a_1] (Device[id = n_0], Device[id = n_1])))

The interpretation states that there is a connection, via a_1, between devices n_0 and n_1. (The identity numbers have been copied into the semantic interpretation in order to enable cross-referencing with the structural description. The rearrangement of the items within the connection has been carried out purely for the sake of consistency with the accepted conventions of semantic representation.)

The semantic description indicates what it *means* to have a network of a given configuration. Hence, the computation of such a description would offer a way of predicting, automatically, what it would mean to reconfigure a particular network in a particular manner.

It should be noted that the complexity of a network representation will depend to some extent upon the level of network-structure at which it is to be considered. At the most abstract level, where we are concerned with the superservices afforded to users, all devices are considered equal and have an ability to communicate with one another, thus forming a fully interconnected mesh. Any realisation of a combination of services will employ a subset of this full interconnection. At a more concrete level, where we are concerned with the physical communication among the networked devices, we have to draw a distinction between *end systems* (such as those devices that communicate directly with the user and the surrounding context) and *core systems* (those devices that make up the network infrastructure). The network viewed at this level comprises less than a full interconnection.

11.3.3 The six-level framework

Having seen how the syntactical and semantic aspects of networks may be dealt with, we are now in a position to return to the question of how the (re)configuration of a network may be viewed as a process involving all six levels of the semiotic framework.

As implied earlier, the social world provides the context within which the need for the network arises. This, then, is where the purpose of commissioning the network, and the motivation behind its specific design, originate.

Given the purpose and motivation, the process begins with developing the network. The process of system development involves establishing the requirements and carrying out the work of design, construction, evaluation, and maintenance. Choosing an initial configuration for the network is part of the design activity, while reconfiguration is part of the maintenance process and amounts to a partial redesign and reconstruction.

These activities of (re)design involve planning how the network will be (re)configured, so that it may function effectively within its (evolving) context of use. Since this activity serves to relate system to context, it is pragmatic in nature. It has the effect of moving forward from the situation where the motivation for a new (or renewed) network exists, on to the activity of actually conceiving the intention or plan to bring that suitably (re)configured network into being. It can, in principle, be carried out either manually or automatically or semi-automatically (with the task shared in one of various possible ways between human and system).

This pragmatic activity involves deploying semiotic resources, namely the semantics and syntactics of network representations, both of which have been outlined above. In this way, the pragmatic activity gives rise to semantic and syntactical activity.

The semantic activity consists in the formulation of the (re)design of the network. This involves deciding on the devices to be incorporated into the network, the properties of those devices and the (re)configuration of those devices.

The syntactical activity consists in expressing the formulation in a form that can be read, whether by a human or a machine or both, as appropriate. The expression may take the form of a diagram or some alternative, for example a purely symbolic description such as (13) above.

The expression of the network (re)design will be manifested in some observable way, whether as a drawing or as an arrangement of legible alphanumeric characters or a combination of the two, and/or as an internal machine-state. This manifestation is an empiric activity, which results in the representation receiving a tangible, observable embodiment on some surface within the physical world and/or in computer hardware storage.

Any system must have the requirements of its users at heart. The user's utilisation of a system (modelled in terms of pragmatic intentions within the context of the social world) will impose requirements upon the kind of network that is necessary and sufficient to support the user's intentions. These requirements will relate to the abstract level of superservice provision, referred to above. An important issue with regard to auto-reconfiguration is how to ensure that the more concrete representation, relating to tangible devices and their physical intercommunication, is capable of underpinning the required abstract, superservice-oriented representation. However, this is too broad a question to pursue here.

11.3.4 Some benefits

The application of semiotic analysis to the field of self-configuring networks has brought two main benefits. Firstly, it has allowed us to maintain, without inconsistency, two different views of the devices and connections within a computer network. Clearly, from one perspective, the network is an assemblage of hardware, and when we consider the communication between user and networked system that is precisely how we view the network. On the other hand, the devices and connections can also be considered as a structure that makes sense, and this is exactly how we view the network when we treat it as an object about which, rather than with which, to communicate.

Secondly, our approach has led to the maintenance of a clear distinction between the different levels within the framework, and in particular between semantics and syntactics. The stratified nature of the approach has encouraged a separation between the structure of a network-representation and its meaning, and straightforwardly accommodates the fact that the same meaning may be expressed in very different (syntactical) forms, such as a diagram or a purely text-based formalism. It also leads naturally to a distinction between (i) the (syntactical) recognition of network plex structures as being either well-formed or ill-formed, (ii) the (semantic) interpretation of those structures to reveal what they mean and (iii) the (pragmatic) understanding of their capabilities in relation to their context of use.

The motivation for adopting a semiotic approach to self-configuring networks is not that we expect it necessarily to lead to improvements of a purely technological nature in respect of such matters as switching, routing, capacity, or efficiency. Rather, it lies in the fact that semiotics offers a framework which accommodates human beings, with their individual needs and desires and their social and organisational goals and aspirations, as well as the technology that they employ, and which makes it possible for all the different aspects of communication involved in the human use of networked systems to be handled in a homogeneous and consistent manner.

11.4 Conclusion

To summarise, it seems that OS does, indeed, lend itself as an appropriate and useful theoretical framework within which to conduct research into self-configuring networks. The particular advantages offered by the application of the six-level model have just been outlined, and they stem from the enhanced analytical understanding of the field of enquiry that the application of the framework has made possible.

We have now established several avenues for further research. Firstly, we need to explore in more detail how our semiotic analysis of network representation may be employed in the automatic (re)configuration of networks of reasonable complexity. Secondly, we need to pursue the question of how the semiotic modelling of intranetwork communication may help implement the intentions of a user requiring networked services, even in the event of breakdowns. Thirdly, in order to accomplish this second objective, we need to propose a means of formally representing user intention. This would then be interpreted by a networked system with the goal of providing services for the user. The representation would be generated by the system on the basis of interaction with the user, perhaps with the aid of machine learning aimed at automatically inferring intention on the basis of previous experience of user behaviour. In this way, our application of semiotics provides us with an engaging research agenda.

References

1. Andersen PB (2002) Pervasive computing and space. In: Liu K, Clarke RJ, Andersen PB, Stamper RK (eds.), Organisational Semiotics. Evolving a Science of Information Systems, Kluwer, Boston, MA, pp. 133–152
2. Brynksov M, Andersen PB (2004) Habitats, activities and signs. In: Filipe J, Liu K (eds.), Organisational Semiotics. Proceedings of the 7th International Workshop on Organisational Semiotics, INSTICC Press, Setúbal, Portugal, pp. 128–141
3. Connolly JH (2002) Accommodating natural language within the organisational semiotic framework. In: Liu K, Clarke RJ, Andersen PB, Stamper RK (eds.), Coordination and Communication Using Signs, Kluwer, Boston, MA, pp. 1–21
4. Connolly JH, Phillips IW (2002) User-system interface design. An organisational semiotic perspective. In: Liu K, Clarke RJ, Andersen PB, Stamper RK (eds.), Organisational Semiotics. Evolving a Science of Information Systems, Kluwer, Boston, MA, pp. 119–132
5. Connolly JH, Phillips IW (2002) Handling complexity in networked multimedia systems. The contribution of semiotics. In: Ragsdell G, West D, Wilby J (eds.), Systems Theory and Practice in the Knowledge Age, Kluwer, New York, pp. 399–405
6. Connolly JH, Phillips IW (2002) Semiotics and the Evaluation of Web-based Communication. Proceedings of the International Conference on Creativity and Cognition 02, Loughborough University, Loughborough, UK, pp. 37–40
7. Connolly JH, Phillips IW (2004) On the semiotic analysis of international communication over computer networks. In: Liu K (ed.), Virtual, Distributed and Flexible Organisations. Studies in Organisational Semiotics, Kluwer, Dordrecht, The Netherlands, pp. 277–291
8. Connolly JH, Phillips IW (2004) On the semiotic analysis of multimedia documents at the level of pragmatics. In: Filipe J, Liu K (eds.), Organisational Semiotics. Proceedings of the 7th International Workshop on Organisational Semiotics, INSTICC Press, Setúbal, Portugal, pp. 12–23
9. Esler M, Hightower J, Anderson T, Borriello G (1999) Next Century Challenges: Data-centric Networking for Invisible Computing. The Portolano Project at the University of Washington. In: Imielinski T, Steenstrup M (eds.), Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MOBICOM 99, Seattle, Washington, USA, August 1999. ACM Press, New York, pp. 256–262
10. Fu KS (1982) Syntactic Pattern Recognition and Applications, Prentice-Hall, Englewood Cliffs, NJ
11. Phillips IW, Connolly JH (2002) The signature of the signature. In: Jorna R, Gazendam H, Cijsouw R (eds.), Proceedings of the Fifth Workshop on Organisational Semiotics, Delft, The Netherlands, pp. 129–143
12. Sousa JP, Garlan D (2002) Aura: an architectural framework for user mobility in ubiquitous computing environments. In: Bosch J, Gentleman M, Hofmeister

C, Kuusela J (eds.), Software Architecture. System Design, Development and Maintenance. Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture, Montreal, Canada. Kluwer, New York, pp. 29–43

13. Stamper R (1991) The semiotic framework for information systems research. In: Nissen HE, Klein HK, Hirschheim R (eds.), Information Systems Research. Contemporary Approaches and Emergent Traditions, North-Holland, Amsterdam, pp. 515–527

Chapter 12
The Semiotics of Usage-Centred Design

J. Ferreira[1], J. Noble[1], R. Biddle[2]

[1]*Victoria University of Wellington, Wellington, New Zealand*
[2]*Human-Oriented Technology Laboratory, Carleton University, Ottawa, Canada*

Abstract

User interface design is a complex activity, which makes it difficult to control and master. A well-designed user interface can be seen as one where designers have correctly mapped the application domain onto the solution domain. This mapping, or bridge, may be helped or hindered by the design methodology and the success of any software engineering methodology depending on how good a bridge it provides between the application domain and the solution domain. A good match between the requirements and the implementation reduces the risk of having to make costly and major changes to the user interface at a late stage in development. Usage-centred design (UsCD) has been a very successful user interface design methodology. Although successful, there is no underlying theory as to *why* this should be the case. This chapter uses semiotics to provide a better understanding of the models and the process of UsCD.

12.1 Introduction

It is generally accepted that user interface design is a challenging and often poorly understood activity [19, 22]. The success of UsCD has been well documented in a number of software projects [4, 13, 27, 33, 35] and is seen by some authors as a methodology that can aid user interface design on agile software development projects [10, 27]. UsCD is a design methodology that

P.-J. Charrel and D. Galarreta (Eds.),
Project Management and Risk Management in Complex Projects, 211–229.

produces a user interface from several derivations of successive abstract models [11]. Constantine talks of the advantage of modelling at the abstract level as providing the user interface designer with more "creative leverage" than a design process that moves to the concrete level very early on [11]. Another important aspect that sets UsCD apart from other design methodologies is the focus on *usage*, i.e. user tasks. We believe that evaluating UsCD from a semiotic perspective can give new insights into the design process and the user interface, which results from that process. This chapter also considers how UsCD relates to Stamper's semiotic ladder as another

> way of revealing how our knowledge of the world (both physical and spiritual) is constituted, and how our awareness of the process is reflected in the dynamics of various semiotic systems or directions [25].

This chapter begins with a brief overview of UsCD in section 2. We then introduce the concepts and definitions of semiotics that are to be applied in section 3. In section 4, we justify the use of semiotics and also discuss the semiotics of each of the UsCD models and highlight the interesting observations. At this point, in section 5, we turn our attention to the semiotic ladder suggested by Stamper and show how the models of UsCD relate to this framework – discussing each level in turn. At the end of section 5, another brief discussion of important points is presented, and finally we conclude in section 6.

12.2 Model Driven Design

According to Arias *et al.*:

> Models are the externalisations that (1) create a record of our mental efforts, one that is ''outside us" rather than vaguely in memory, and (2) represent artefacts that can talk back to us and form the basis for critique and negotiations [5].

These models are the externalisations, or descriptions, of processes in the domain [2]. As in object-oriented design, we will investigate the models of UsCD with respect to two domains that require modelling – the *application domain* and the *solution domain*, adopting Bruegge and Dutoit's [7] definition of these two domains. The application domain represents all aspects of the user's problem and the solution domain is the space containing all possible implementations.

Turning our attention to user interfaces, Biddle, Constantine, and Noble propose models as ideal tools for answering questions like:

> What capabilities must be present in the user interface to solve the users' problems? How should they be organised into handy collections? How should the work flow within and between the various parts of the interface? [6]

UsCD makes heavy use of modelling in an attempt to integrate *usability* into the design of the user interface. Usability is a well recognised essential characteristic of any interface, and its incorporation into the design can ensure that resources are not wasted by developing user interfaces that have to be significantly altered at a later stage. Constantine and Lockwood have written a very comprehensive book on the UsCD process [12]. Due to space constraints, we focus on its three core models:

1. Role model – the relationships between users and the system
2. Task model – the structure of tasks that users will need to accomplish
3. Content model – the tools and materials to be supplied by the user interface

UsCD can be described as understanding and skills that are successively translated and derived through the use of models, since each model builds on the previous one. These will be discussed in the next sections, along with the canonical abstract prototype (CAP) – which is the last step of the UsCD process before the user interface or realistic prototype is implemented.

12.3 Semiotic Approaches

The semiotic explanations in this chapter are based on the notion of sign, as posited by the American philosopher Charles Sanders Peirce. Noth [26] and Chandler [8] give excellent introductions to Peircean semiotics, so we will only supply a basic introduction here. In section 5 we go on to examine the models of UsCD in terms of the semiotic ladder suggested by Ronald Stamper [32], which we introduce in this section.

12.3.1 Peirce

To Peirce, a sign is "something which stands to somebody for something in some respect or capacity." [28] Peirce's model of the sign consists of a triadic relationship containing three parts: the *representamen*, the *object*, and the *interpretant*. The representamen stands to somebody for something in some respect or capacity. It addresses somebody and creates in the mind of that person an equivalent, or perhaps more developed sign. The object is the actual thing the sign stands for [28]. The interpretant is therefore the sign created in the mind of the perceiver or the reaction caused by the object in the perceiver [3]. Peirce classified signs based on the relationship between the object and representamen. The three fundamental sign categories he described are iconic signs, indexical signs and symbolic signs. If the representamen resembles, or in some way imitates the object, then the sign can be seen as an iconic sign. Indexical signs exist because we can infer or observe a link between the representamen and object and in this case the sign does not represent its object. If the relationship between the object and the representamen is a purely conventional one that must be learned by the perceiver, then the sign is symbolic.

These sign categories are not mutually exclusive – most signs contain elements of iconicity, indexicality, and symbolism in varying measures. It is very rare, and some argue impossible, to find signs in the real world that belong to solely one category.

12.3.2 Stamper

The fundamental concept for Stamper's semiotic ladder is the sign; however, Stamper modified the Peircean definition of the sign to "something which stands to somebody for something in some respect or capacity, in some community or social context." [31] This new definition seems more appropriate when talking about signs that are used among a certain group of people, for example, a team of user interface designers. The consequence of Stamper's definition is that a sign may only be understood within this group and be less understandable (if at all) to those outside the group. Using Stamper's semiotic ladder (explained in section 5), we can focus on different aspects of signs ranging from their physical appearance to their social consequences [1]. The six specific levels on which Stamper proposes signs examined are: the social, pragmatic, semantic, syntactic, empiric, and physical levels. We will explain these in more detail with respect to the models of UsCD in section 5.

12.4 Models as Signs

Andersen and Nowack explain a triadic concept of models as signs: The model represented by the special notation that inherently belongs to it, and renders it visually is the *representamen*. This representamen refers to some other tangible thing, which is the concept the model represents. The *interpretant* is what can be called a *referent system*. This referent system is "the particular way we choose to relate the model to reality." [3] Hence, each model will have a different referent system as its interpretant, since each model employed by a process such as UsCD is intended to model a unique aspect of a domain. UsCD designers should agree on what the referent system of each model is so that effective communication of the models can take place. The *object* of the model as sign is then the phenomenon it is intended to represent. Stamper's definition of the sign, as something that stands for something else for some community [31], is particularly appropriate in this context: community referring to all those involved in the design process of a specific system.

Closely following Andersen and Nowack's semiotic analysis of object-oriented models [3], we realise that the UsCD models can be related to Peirce's triadic model of the sign where the actual models that are constructed during the user interface design process become the representamens that stand for concepts in the underlying domain. The way the designers interpret the model is the interpretant, and the object relates to the corresponding real world phenomenon in the domain.

Figure 1 shows an example of the Peircean triad applied to the User Role Model of UsCD, where the representamen is the visual rendering of the User Role Model.[1] The object is the concepts which the User Role Model is intended to represent, namely the collection of user roles and the relationships that exist between them. The interpretant is the referent system that the user interface designers have agreed upon. In this instance, they have agreed that the User Role Model stands for the collection of user roles and their inter-relationships. Note that the relationship between the object and representamen is of greatest interest in this chapter, and it becomes clear that knowing what domain the model is describing helps to pin down the object of the model as sign.

Here it is also worth noting that all models employed by a user interface design process, as a collection, can be seen as an indexical sign. That is to say that the models created by user interface designers, combine to form a sign that indicates that a user interface is under construction, in a similar way that smoke is an indexical sign of fire.

Throughout the rest of this chapter, we will refer to an example of the design of a user interface to a computer system for use in a small pizza business. This example was developed as part of a course in User Interface Design,[2] teaching the methods of the UsCD process. The models were produced as part of the set tasks that were carried out as group work.

---

[1] The user role model is a combination of the structured role models and the user role map as explained in section 4.1

[2] COMP311: user interface design, Victoria University of Wellington, New Zealand (2004)

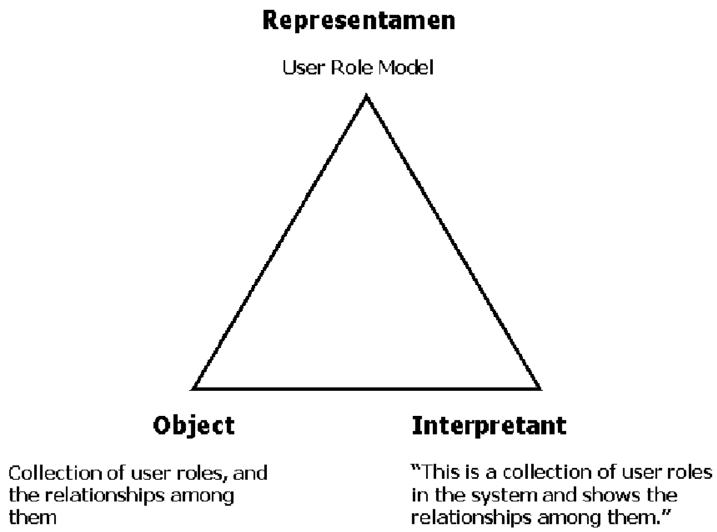

Fig. 1. A diagram of the Peircean triad as applied to the User Role Model

12.4.1 User role model as a symbolic sign of the application domain

Constantine and Lockwood define a *user role* as "an abstract collection of needs, interests, expectations, behaviours, and responsibilities characterising a relationship between a class or kind of users and a system." [12] The important idea is that a role is not restricted to only one person. Many roles can be played by one person and one role can belong to many people who are involved in the interaction with the system. The User Role Model is a combination of Structured Role Models (Fig. 2) and a User Role Map (Fig. 3). The Structured Role Models are profiles of the user roles as they exist in the application domain, and provide information such as the level of proficiency and specific usability criteria for a given role. The User Role Map simply lists the various roles in the system and how they relate to each other. Therefore, we can say that the User Role Model involves identifying who the most important users of the system are and what tasks they will be performing with the system. Identification of the user roles and the relationships between them, guides decisions about what functionality will be implemented in the interface. The most influential roles, called *focal roles*, are implemented first in the user interface before the designers turn their attention to other possible roles.

Order Taker	
Purpose	Order Taker is a pizza shop employee that comes into contactwith the customers. His/her duties are: 1. talking to customers 2. offering menu items 3. taking orders 4. confirming orders 5. taking payment 6. querying and modifying orders
Domain knowledge	Needs to have a good knowledge of the menu content and prices of its items, as well as time frames needed for cooking and delivery.
System Knowledge	The main interaction of Order Taker with the system is through order details processing and customer details processing. Therefore they need a good knowledge of these two processes. The adequate training will be provided.
Background	According to the above two sections, Order Taker will have pizza and previous system knowledge. In addition he/she will have good customer service skills.
Proficiency	Order Taker must have a level of proficiency with the system.
Interaction	The interaction with the system is very frequent and high intensity,
Information Flow	Information originates from the Customer. The volume and complexity is low, as the Order Taker only handles one order at a time.
Usability Priorities	The system must be easy to learn, tasks should be highly memorable and accuracy is very important.

Fig. 2. Structured Role Model

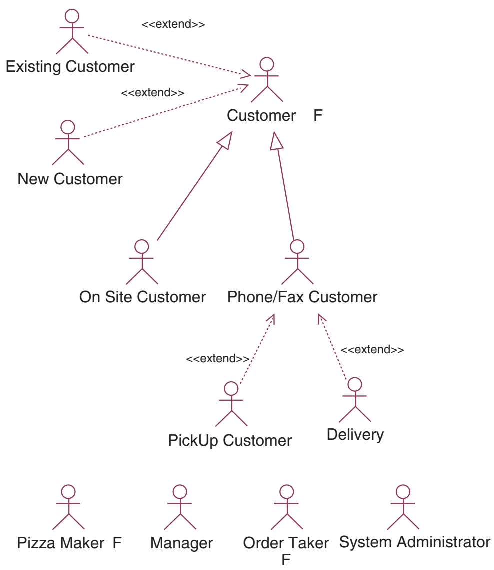

Fig. 3. User role map

In our example of the pizza business, we identified our roles as those appearing in Fig. 3. The most influential roles and those who would drive the rest of the design process were identified as the customer, the order taker and the pizza maker. According to the role map there is only one kind of pizza maker and order taker, but several kinds of customer.

The User Role Model is a mainly symbolic sign. The User Role Map component represents the roles and their relationships through the conventional notation of stick men for roles and arrows to indicate relationships. These conventions have to be learned and are not immediately obvious to a perceiver who is not familiar with UsCD. The Structured Role Model is a verbal description of a role and is therefore also a symbolic sign.

12.4.2 Task model as a symbolic sign spanning the application and solution domains

The User Role Model, as discussed above, is the solution that becomes the requirements for the Task Model. The Task Model is a representation of both *what* and *how* user tasks are performed, and consists of *essential use cases* (Fig. 4) and a Use Case Map. Use cases are a common technique in software engineering, but it is appropriate to give Constantine and Lockwood's specific definition here, as it differs slightly from the traditional definition: "An essential use case is a structured narrative, expressed in the language of the application domain and of users, comprising a simplified, generalised, abstract, technology-free, and implementation-independent description of one task or interaction that is complete, meaningful, and well-defined from the point of view of users in some role or roles in relation to a system and that embodies the purpose or intentions underlying the interaction." [12] The Use Case Map, in a similar way to the User Role Map, shows a list of all the essential use cases[3] and the relationships that exist among them.

Returning to the running example, for brevity, we will focus on one user task required to be supported by the pizza business's user interface. We can choose any task belonging to the order taker, pizza maker or customer role, since all three are focal roles. Our example of an essential use case in Fig. 4, is about the payment use case describing the order taker's task of handling the payment of an order.

As with the User Role Model, the Task Model is a mainly symbolic sign, since the notation used in the Use Case Map bears no resemblance to

---

[3] From this point on we will refer to *essential use cases* as *use cases* for matters of convenience.

PAYMENT <<extends>> ENTER ORDER *User Intention*	 *System Responsibility*
  2. Select Payment Type 3. If the payment type* is account then select customer account 4. Confirm selections	1. Display Payment Dialogue 5. If payment type is account, debit account 6. If the payment type is deferred, mark order as unpaid, otherwise record payment and print receipt
Note: *Payment type may be: cash, credit card, cheque, eftpos, deferred or account.	

Fig. 4. Essential use case

the set of tasks users will be performing with the user interface. Neither do the use cases, a "structured narrative", resemble a user task. What we do find in the use case, to a lesser degree, is some properties of a degenerate index. A degenerate index is not a causal effect of the object, as smoke is a causal effect of fire, but "a device which enables the interpreter to place himself in direct experiential or other connection with the thing meant." [23] This is exactly what a use case allows designers as well as users to do: experience the system at each task. A use case is degenerate in that it consists of a process of firstness [34], i.e. the tasks are modelled in the use case in such a way that there is a likeness between the tasks in the use case and the tasks as they will be performed with the future user interface. The likeness, however, is not in appearance but in "respect to the relations of their parts." [28]

The use case is further also a model of two different domains – that of the application domain and of the solution domain. The user intentions belong to the application domain, since these are the tasks the user is required to perform in the application domain. Traditionally, use cases are seen as models of the application domain [15], but we propose that the system responsibilities belong to the solution domain, seeing that these responsibilities are necessarily inherent of the system that will be "selected" from all the possible implementations of the solution domain.

12.4.3 Content model as indexical sign of the solution domain

"Content models represent the contents of user interfaces and their various constituent sections or parts independent of details of appearance and behaviour." [11] This model involves identifying all the *materials* and *tools* that a user will require in the interface. Materials are the things users want to see and manipulate, and tools enable users to do things with the materials [12]. The Content Model consists of *interaction contexts* that are identified by grouping together those tools and materials needed for carrying out some particular task or related tasks and the use cases are the source from which the designers determine what tools and materials the users will require to perform these tasks. Different interaction contexts will require different tools and materials. Representing the tools and materials with stick notes, and the interaction contexts with sheets of paper, allows designers to organise user interface features and make changes quickly and cheaply. Also included in the Content Model is the navigation map that shows the interconnection between the interaction contexts and the circumstances under which the interaction context changes.

In the process of designing the pizza business's user interface, this artefact was not expected to remain a permanent model to be used for future reference like the previous models we have discussed. In our group experience, the content inventory served the function of a "throwaway prototype of the prototype." It allowed for the quick and cheap exploration of possible prototype designs and was discarded when the final prototype design had been decided on.

There is a large element of indexicality in the Content Model. As Peirce has noted, the indexical sign "is a sign of its object by virtue of being connected with it" [28] and this connection is the behaviour of the future user interface that we model with the Content Model. The dynamic aspect of the Content Model, i.e. the change from one interaction context to another, creates in the mind of the perceiver, a link with the behaviour that will exist in the future user interface. The Content Model, like the Task Model allows the user to experience some aspects of the behaviour of the future user interface it is referring to, without the perceiver having physical contact with the future user interface.

The Content Model clearly has symbolic elements as well – the sticky notes, paper, and text of the Content Model do not resemble the future user interface in any way.

The underlying domain of the Content Model is the solution domain, and hence, this model is an indexical sign of the solution domain. This model is also the first UsCD model that is representing the future user interface in a more direct way, i.e. it is the first model that has the future user interface as its object. The next step is to design the CAP so that the designers can come closer to an actual interface in the solution domain.

12.4.4 Canonical abstract prototype as indexical sign of the solution domain

With the organisation and presentation of functionality mostly determined in the Content Model, designers next refine the appearance and behaviour of the user interface using the CAP. The CAP models the various interactive functions needed within the realised user interface [11] that will become visible features once implemented. This is done using a standardised notation called *canonical abstract components* – a kind of shorthand for interactive functions. Refer to Constantine *et al.* [14] for the current set of canonical abstract components.

The pizza business's interface required a way of recording payments for orders. The payment use case (Fig. 4) contains all the requirements that would be needed to design some kind of interface element that would have the required features. From inspecting the use case it was clear that a user would need a tool for selecting the type of payment that the customer presented, a material to display the amount owed by the customer, an active selectable collection in order to select the customer's account from a list of customer accounts kept in a database, a tool to exit (corresponding to the delete/erase tool [14]) and finally, a select tool that would confirm any choices made during the payment use case. Figure 5(a) is the CAP of the payment dialogue.

The canonical abstract components correspond to the features that will be present in the future user interface and a link is created in the mind of the perceiver between these components and future implemented user interface components. The canonical abstract components are, therefore, indexical signs. Similar to a geographical map, for instance, the CAP specifies the layout of user interface features and this further strengthens the indexicality of the CAP as sign.

Since the CAP provides more visual information about the size and organisation of user interface features, it is much closer in appearance to how the resulting interface will look once implemented than the Content Model, and also a much stronger iconic sign. Precise details such as colour and borders still remain undetermined, so it is not yet fully iconic.

Once the CAP has been designed, a realistic prototype can be developed – which is iconic in the same way that a thumbnail is an iconic representation of a photograph – and subsequently the actual user interface. Like the Content Model, the CAP is a model of a possible implementation in the solution domain.

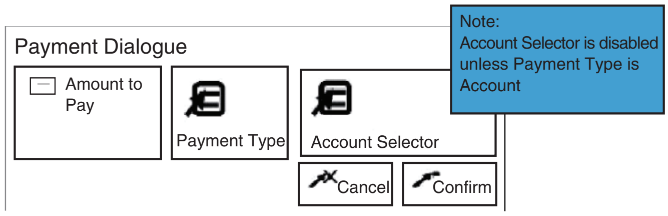

Fig. 5(a). Canonical Abstract Prototype

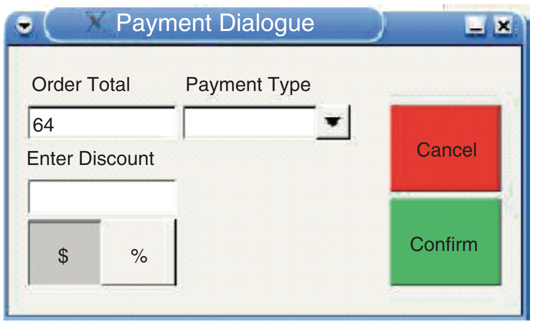

Fig. 5(b). Implementation of the Canonical Abstract Prototype in Fig. 5(a)

The implementation of the CAP in Fig. 5(a) resulted in the payment dialogue in Fig. 5(b).

12.4.5 Discussion

Interestingly, the models represent semiotic transformations from the symbolic to the indexical to the eventual iconic representation of the interface. These semiotic transformations occur in the opposite direction to what Peirce has referred to as a "regular progression" [28] that is, a progression of signs from iconic to indexical to symbolic. However, the transitions from symbolic to indexical to iconic signs are moving the designers closer and closer to the actual form the user interface will take – the necessary order, it seems, in order to home in on the user interface that implements all the functionality required by the users.

Another feature of these successive transformations is that they carry design information from one model to the next – information and data flows from the User Role Model to the Task Model, then from the Task Model to the Content Model, then from the Content Model to the CAP and then from the CAP to the prototype or final implementation of the user interface. This phenomenon brings to mind Eco's concept of finite unlimited semiosis. But where Eco has agreement on the meaning of metaphor in mind [29], user interface designers have the user interface in mind, and their eventual agreement results in the implemented interface. In relation to Peirce's model – the Task Model becomes the effect of the action taken by the interpreter (designer) on perceiving the representamen of the role model; the Content Model becomes the effect of the action taken by the interpreter (designer) on perceiving the representamen of the Task Model, and so on until the implemented interface is reached. Eco succinctly describes unlimited semiosis as a process where

> ... the repeated action responding to a given sign becomes in its turn a new sign, the representamen of a law interpreting the former sign and giving rise to new processes of interpretation. [16]

The fact that each model successively builds on the previous one allows for traceability – not only in the completeness of the models at each step but also in the ability to recall decisions made at each stage of the development process. Requirements traceability is especially a concern and potential risk of software engineering projects [20].

12.5 Usage-Centred Design and the Semiotic Ladder

We now turn to the semiotic ladder of Stamper to continue our discussion on the models of UsCD. Although Stamper's semiotic ladder [32] (see Fig. 6) was proposed as part of his endeavour to define information, communication and meaning, we can apply this ladder to the user interface and consider what aspects of the user interface is modelled by UsCD at each level.

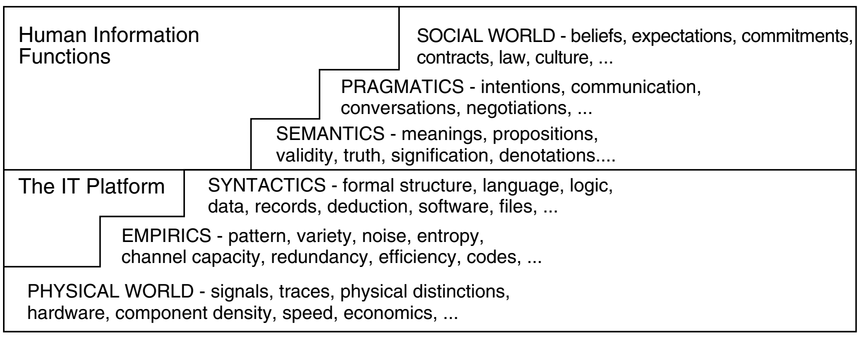

Fig. 6. Stamper's semiotic ladder

12.5.1 Social level

The FRISCO Report [18] states that recognition of the social dimension is essential in understanding the properties and purposes of signs. The social level may refer to the design environment, where designers use the models of UsCD to communicate their designs to each other and to the users, and also to the context in which the user interface is to be deployed. Just as Peirce noted that a sign is not a sign unless interpreted as such, the models of UsCD have no significant meaning outside of the context in which UsCD is the design methodology. It is only when the user interface designers and the potential users agree on the properties and purpose of the models, that they become meaningful. Recalling Stamper's definition of the sign, the community or social context now refers to the group of user interface designers employing UsCD and the potential users and operational environment of the resulting interface. At the social level, UsCD provides us with the User Role Model for modelling the roles of the potential users of the interface to be designed, while there is no explicit model that deals with organisational and cultural rules or institutional policies of the environment in which the user interface is to be deployed.

12.5.2 Pragmatic level

Pragmatics is concerned with the *usage* of signs. Morris points out that at this level we study the relation between signs and their users [24]. Connolly and Phillips recognise the interface as an "instrument of user-system communication" [9], and therefore, we must examine the UsCD models that deal with the relationship between the user and the system. As discussed above, the User Role and Task Models are tools for examining what a user interface requires for it to be useful to its users. These two models describe the usage aspect of the user interface and helps designers decide on the functionality of the interface, or the "meaning" of the interface as complex sign. The fact that designers decide on this "meaning" (the object of the sign in Peircean terms), implies that the user interface is an *intentional* sign, i.e. the communication of the functionality of the interface is intentional as opposed to unintentional or natural.

The *effect* of the User Role and Task Models will depend on the social level: the user roles and the relationships between them, as well as the tasks they need to accomplish with the system will, through the User Role and Task Models, be interpreted by the user interface designers at the social level in order to proceed to deciding on the contents of the user interface, and hence, the Content Model.

12.5.3 Semantic level

At the semantic level, the behaviour and validity of the user interface needs to be verified. This is already partly accomplished through the creation of the User Role and Task Models, which model the usage of the interface. Another useful model for representing the semantic meaning of the user interface under development is the Content Model, which specifies how the interface will function and how it is organised as a whole. The validity of all three models is verified when the users accept that these models correctly represent what the system should do and that they include all possible roles and tasks.

12.5.4 Syntactic level

The syntactic level applies to the structure of signs and the rules that govern the way they are combined. In the case of the user interface, at the lowest level is the program code that has generated that interface. This code belongs to the syntactic level in that the programming language in which it is written prescribes the rules for producing valid code, and hence, producing the user interface. At a higher level, we may see user interface syntax as the steps a user has to complete in order to achieve a certain goal [17]. Here again the Task Model would be of most use in determining what steps are required to complete a task or achieve a goal. UsCD has no model for representing user interface code. However, the CAP, with its syntax of canonical abstract components, can be seen as a syntactical representation of what the eventual user interface is expected to look like. Although the rules for combining these canonical abstract components are not fixed, and their combination may vary from user interface to user interface, they are the closest representation of the syntactic meaning of the user interface under development. The person responsible for writing the code for the user interface will still be relied upon to translate the CAP into the appropriate program code.

12.5.5 Empirical level

Detectable patterns and variety through the use of statistics belongs to the empirical level. In a user interface it is possible to detect the various shapes of features and the number of times they are repeated. In the UsCD process, we can determine such statistics as the number of roles in the User Role Model, the number of tasks in the Task Model and the CAP may show detectable patterns and variety in the components of the user interface to be designed. However, this information will not be fully applicable to the resulting user interface until it has actually been implemented.

12.5.6 Physical level

The physical level of a user interface consists of the group of pixels that render it on a computer monitor. UsCD has no model to offer because if it had, then this would be the implemented user interface, or a realistic prototype of the user interface.

12.5.7 Discussion

Not surprisingly, the modelling in UsCD has a strong focus on the pragmatic aspect of user interface design. The success of user interface design hinges on the designer's ability to make it possible for users to interpret their signs. Not only the discrete signs that the user interface consist of, but also the user interface as a complex sign in itself. It is very important that the models which determine the intention of the user interface signs allow the user to interpret those intentions. When a user clicks on an icon of a printer and their file is erased, then there is an obvious misalignment of the user's intention and the designer's intention.

We mentioned in section 2 that the UsCD models undergo several translations during the course of the design process. Applying the concepts of the semiotic ladder, it becomes clear that the successful translations of the models depend heavily on the social level, specifically on norms such as expertise of analysis, design, and testing [21], as well as knowledge of the UsCD process itself. The impact of a weakness at the social level, such as lack of expertise, could be a dramatic one and result in a user interface that does not meet the requirements of the users.

12.6 Conclusion

Approaching UsCD from a semiotic perspective has led to some useful insights into the process and its models. Applying Peircean semiotics, we observed that the semiotic transformations occur in the opposite direction to what Peirce referred to as a "natural progression." The successive models provide traceability between the various stages of design and mitigates the risk of developing extraneous features or of omitting necessary ones.

Examining UsCD in terms of Stamper's semiotic ladder, highlighted that UsCD models focus strongly on the pragmatic aspect of the user interface and that the success of the process depends heavily on the social level to develop a user interface that is a close match with its requirements.

References

1. Ågerfalk PJ, Karlsson F, Hjalmarsson A (2002) Exploring the explanatory power of actability – the case of Internet-based software artefacts. In: Liu K, Clarke R, Andersen P, Stamper R (eds.), Proceedings of Organizational Semiotics: Evolving a Science of Information Systems, IFIP TC8/WG8.1 Working Conference on Organizational Semiotics: Evolving a Science of Information Systems, Montréal, Canada. Kluwer Academic Publishers, Norwell, MA, pp. 1–20
2. Andersen PB, Hasle P, Brandt P (1997) Machine semiosis. In: Posner R, Robering K, Sebeok TA (eds.), Semiotics: A Handbook About the Sign-Theoretic Foundations of Nature and Culture. Walter de Gruyter, Berlin, Germany, pp. 548–571
3. Andersen PB, Nowack P (2002) Tangible objects: connecting informational and physical space. In: Qvortrup L (ed.), Virtual Space: Spatiality in Virtual Inhabited 3D Worlds, Springer, London, pp. 190–210
4. Anderson J, Fleek F, Garrity K, Drake F (2001) Integrating Usability Techniques into Software Development. IEEE Software 18(1), pp. 46–53
5. Arias E, Eden H, Fischer G, Gorman A, Scharff E (2000) Transcending the individual human mind – creating shared understanding through collaborative design. ACM Transactions on Computer-Human Interaction 7(1), pp. 84–113
6. Biddle R, Constantine L, Noble J (2003) Usage-Centered Design and Software Engineering: Models for Integration, Proceedings IFIP Working Group 2.7/13.4, ICSE 2003 Workshop on Bridging the Gap Between Software Engineering and Human-Computer Interaction, Portland, Oregon, pp. 106–113
7. Bruegge B, Dutoit A (1999) Object-Oriented Software Engineering: Conquering Complex and Changing Systems, Prentice-Hall PTR, Upper Saddle River, NJ
8. Chandler D (2002) Semiotics: The Basics, Routledge, London
9. Connolly J, Phillips I (2002) User-system interface design. In: Liu K, Clarke R, Andersen P, Stamper R (eds.), Proceedings of Organizational Semiotics: Evolving a Science of Information Systems, IFIP TC8/WG8.1 Working Conference on Organizational Semiotics: Evolving a Science of Information Systems, Montréal, Canada. Kluwer Academic Publishers, Norwell, MA, pp. 119–132
10. Constantine L (2002) Process Agility and Software Usability: Toward Lightweight Usage-Centered Design. Reprinted from Information Age 8(8). In: URL http://www.foruse.com/articles/agiledesign.pdf

11. Constantine L (2003) Canonical abstract prototypes for abstract visual and interaction design. In: Jorge J, Nunes N, e Cunha J (eds.), Proceedings of the Tenth International Workshop on Interactive Systems: Design, Specification and Verification, DSV-IS 2003, Funchal, Madeira. Lecture Notes in Computer Science, Vol. 2844, Springer, Berlin
12. Constantine L, Lockwood L (1999) Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design, ACM Press, New York
13. Constantine L, Lockwood L (2002) Usage-Centered Engineering for Web Applications. IEEE Software 19(2), pp. 45–50
14. Constantine L, Windl L, Noble J, Lockwood L (2003) From Abstraction to Realization: Abstract Prototypes Based on Canonical Components – REVISED. In: http://www.foruse.com/articles/canonical.html
15. Düwel S, Hesse W (2000) Bridging the gap between use case analysis and class structure design by formal concept analysis. In: Ebert J, Frank U (eds.), Proceedings of Modellierung 2000: Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik, Fölbach-Verlag, Koblenz, Germany, pp. 143–150
16. Eco U (1984) The Role of the Reader: Explorations in the Semiotics of Texts, Indiana University Press, Bloomington
17. Elwert T (1996) Continuous and Explicit Dialogue Modelling. In: Proceedings CHI'96: Conference Companion on Human Factors in Computing Systems, Vancouver, Canada. ACM Press, New York, pp. 265–266
18. Falkenberg E, Hesse W, Lindgreen P, Nilsson B, Oei J, Rolland C, Stamper R, Van Assche F, Verrijn-Stuart A, Voss K (2000) FRISCO – A Framework of Information System Concepts – the FRISCO Report. IFIP WG 8.1 In: URL ftp://ftp.leidenuniv.nl/pub/rel/fri-fill.zip
19. Galitz W (2002) The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, 2nd edn., Wiley, New York
20. Gotel O, Finkelstein A (1994) An Analysis of the Requirements Traceability Problem, Proceedings of the First International Conference on Requirements Engineering, Colorado Springs, CO, pp. 94–101
21. Liu K, Liao S, Chong S (2002) Semiotics for information systems engineering – reuse of high-level artefacts. In: Carrasco RA (ed.), Proceedings of the Third International Symposium on Communication Systems Networks and Digital Signal Processing, Sheffield Hallam University Press, Staffordshire, UK
22. Maguire M (1999) A review of user-interface design guidelines for public information kiosk systems. International Journal Human-Computer Studies 50(3), pp. 263–286
23. Martinovsky B (1995) Shifting worlds or deictic signs in WWW. Online Publication. In: URL http://www.ling.gu.se/biljana/web.html, SKI/GU
24. Morris C (1938) Foundations of the Theory of Signs, University of Chicago Press, IL
25. Nadin M (1993) Semiotics in action: the pragmatic level. In: Stachowiak H (ed.), Pragmatic, 4, pp. 219–250
26. Noth W (1995) Handbook of Semiotics, Indiana University Press, Bloomington, IN

27. Patton J (2002) Extreme design: usage-centered design in XP and agile development. In: Constantine L (ed.), for USE2002, Proceedings of the First International Conference on Usage-Centered, Task-Centered and Performance-Centered Design, Portsmouth, NH. Ampersand Press, Rowley, MA
28. Peirce CS (1931–58) Collected Writings (8 Vols). Hartshorne C, Weiss P, Burks AW (eds.), Harvard University Press, Cambridge, MA
29. Simpkins S (last accessed 10 June 2005) Critical Semiotics, Lecture 6: Finite infinite semiosis. In: http://www.chass.utoronto.ca/epc/srb/cyber/sim6.html
30. Stamper R (1991) The semiotic framework for information systems research. In: Nissen H, Klein H, Hirschheim R (eds.), Information Systems Research: Contemporary Approaches and Emergent Traditions. North Holland, Amsterdam, The Netherlands, pp. 515–517
31. Stamper R (1996) Signs, information, norms and systems. In: Holmqvist P, Andersen P, Klein H, Posner R (eds.), Signs of Work: Semiotics and Information Processing in Organisations. Walter de Gruyter, New York, pp. 349–397
32. Stamper R (2001) Organisational semiotics: Informatics without the computer? In: Liu K, Clarke R, Andersen P, Stamper R (eds.), Information, Organisation and Technology: Studies in Organisational Semiotics, Kluwer Academic, Boston, MA, pp. 115–171
33. Strope J (2002) Putting usage-centered design to work: clinical applications. In: Constantine L (ed.), for USE2002, Proceedings of the First International Conference on Usage-Centered, Task-Centered and Performance-Centered Design, Portsmouth, NH. Ampersand Press, Rowley, MA
34. Taborsky E (2000) The Complex Information Process. Entropy 2, pp. 81–97
35. Windl H (2002) Designing a winner: creating STEP 7 Lite with usage-centred design. In: Constantine L (ed.), for USE2002, Proceedings of the First International Conference on Usage-Centered, Task-Centered and Performance-Centered Design, Portsmouth, NH. Ampersand Press, Rowley, MA