Institut für Informatik
der Technischen Universität München

A Methodology to Configure IT-Collaboration-Platforms Based on Fuzzy Logic

Céline Laurent

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Jürgen Schmidhuber

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Dr. h. c. Manfred Broy
2. Univ.-Prof. Michael Beetz, Ph. D.

Die Dissertation wurde am 23.10.2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 10.03.2008 angenommen.

Abstract

Today's new engineering processes connect more strongly all the various disciplines together, integrated at the process level as well as through the support of communication and flow of information. Moreover, globalisation has created a need to collaborate and compete with counterparts located thousands of miles away. Bridging the geographical and organisational distances between the teams involved in such projects, especially across several time zones, requires additional activities and efforts.

These additional efforts translate to a substantial planning, coordination and control overhead in the daily work of projects. Some of these challenges can be solved by or with the help of sophisticated tools and platforms. The main focus of IT-systems for distributed and interdisciplinary engineering projects lies on topics like communication, transfer of information, team coordination as well as special cases of the engineering process. The goal is to provide tools and methods so that a geographically distributed and interdisciplinary team can collaborate as easily as co-located teams of specialists in the same domain.

There is already a variety of tools and technologies prevalent that facilitate communication and collaboration. However, choosing the most appropriate IT-support or IT-collaboration-platform is quite difficult, since technology is always in evolution and defining precise requirements on collaboration software is extremely complex.

In this thesis we present a methodology to determine systematically the IT-tools appropriate to specific engineering projects with cross-domain and cross-enterprise characters. In contrast to other approaches, the developed methodology is independent from the processes followed in the project, as well as from the tasks to be executed, and thus, can be applied to each kind of engineering project and combinations of them. It can also be applied to projects which do not have any precisely defined and implemented processes as in the case of innovative engineering projects.

The new methodology uses influencing factors of cross-domain and cross-enterprise projects defining the cooperation form of the project, as well as its profile and context, and then formalise them with the help of fuzzy logic, because these factors are human factors and are often given by in vague terms, which make their definition a big challenge.

To make a decision about the appropriate IT-tools we have correlated the factors to a repertoire of IT-tools, i.e. standard IT-tools supporting collaboration processes, with the help of fuzzy logic. The rules between the tools and the influencing factors are based on an empirical research carried out on our own by experts of cooperation projects, as well as on the prioritisation of the influencing factors resulting from a sensitivity analysis. The method to configure IT-collaboration-platforms has then been used to develop a small software tool: the Collaboration-Platform Configurator (CPC). It gives the project leader an advice on how his collaboration-platform should be configured, i.e. which IT-tools are appropriate or not for his project. However, the method neither provides advice nor guidelines for implementing the proposed solution, nor an explanation why an IT-tool is recommended or not.

The method has then been verified by applying it to existing engineering projects. With these case studies we principally want to verify the method, show its applicability to real projects, and highlight some advantages and disadvantages of this method. Finally, the generated configuration has been implemented with the help of an IT-platform existing at BMW.

The evaluation of the method shows that the resulting collaboration-platforms are optimised to the projects' needs and follow the current business and engineering processes. Using the methodology to configure a collaboration-platform is of great significance to reduce "time to decision" processes in the early phase of cooperation projects, especially if the project must start quickly and the managers do not have time to wait for an analysis of the project's configuration and possible IT-support. Since it reduces the development and implementation time of the platform within an engineering project, taking the methodology results in drastic time-saving and a quicker estimate of the global IT-costs.

Acknowledgements

First and foremost my sincere thanks go to Markus Schichtel, who gave me the opportunity to do the research which resulted in this thesis. His encouragement, guidance, and advice during the entire project greatly supported me on a professional level as well as on a personal one.

My thanks also go to Manfred Broy, who, when asked, immediately agreed to be the mentor of my dissertation even though my topic diverges some from his main research interests. He offered keen insights and very useful feedback on earlier drafts of this document; I profited from as well as extremely enjoyed the various discussions we had. I would also like to thank my co-supervisor, Michael Beetz, for his valuable criticisms.

I am most grateful to the members of my business unit at BMW for contributing toward and maintaining an excellent atmosphere within the project on a daily basis; in particular, many thanks are due to Markus Schichtel, Andreas Hagengruber, Birgit Snowdon, Markus Fricke, and Wilhelm Kerschbaum for their support. Special thanks also go to Walter Trapa who was a never-failing source of encouraging comments as well as to Rainer Fladung who read substantial parts of the thesis and responded with very helpful remarks.

I am especially indebted to Stefan Wagner and Sebastian Winter. They patiently read almost the entire thesis, discussed its contents with me, and proposed several changes; I am obliged to them for this as well as for their continual willingness to listen to and respond to the many questions I had. Furthermore, I am grateful to Marcia Winter who took the time to review my thesis and offered linguistic inputs.

I am also grateful to Silke Müller, Florian Deißenböck, Martin Feilkas, Patrick Keil, Bastian Best, as well as all of the other members of our research group for their active exchange of ideas in work-related areas as well as for pleasant diversions on many occasions.

My parents, Corinne and Jean Laurent, have always lovingly supported, challenged, and encouraged me in my studies and chosen goals. I am profoundly grateful for all you have done for me!

Last, but not least, I would like to thank my sister, Sophie, and my brother, Jérôme, who have always backed me with their emotional support, encouragement, and motivation.

Index of Contents

1. INTRODUCTION

1.1. Motivation

In order to remain competitive, engineering must evolve constantly: not only every engineering domain in itself, but also the interacting on of each with the other. In fact, advances are being made to an increasing degree at the interfaces of different disciplines. Globalisation and the notion of ‘interdisciplinary engineering’, i.e., a new understanding of the processes involved in design and engineering as well as the way in which they interact [Encarnação, 2005], mark the current trend in engineering fields.

1.1.1. Globalisation

Globalisation implies a strong networked and distributed product-development process: it offers vast potential for businesses due to its innovation ability, flexibility, cost reduction, time reduction, and improving quality, while simultaneously it still remains a challenge to use this potential nowadays.

For engineering, as a cooperative and information-generating process, there is an especially increasing pressure that results from globalisation due to intensive, global competition, distributed and challenging markets, as well as technology cycles that are getting shorter and shorter (cf. [Smith et al., 1991], [Wildemann, 1999], [Wheelright et al., 1992] and [Milberg et al., 1996]). Concomitantly, clearly defined processes provide globalisation with possibilities to optimize distribution, networking and the integration of development resources (cf. [Picot et al., 1998]), which makes it possible for enterprises to withstand the competitive pressure.

The challenges that follow from the processes of globalisation become apparent by looking at the example of global networking in the automotive industry. Globally-distributed cooperation between different domains is in effect a fact of daily work. Types of cooperation span from an in-house to a general-enterprise cooperation, from conception to production, in the form of joint ventures, technology and production agreement, to exchange over locations and time zones. The variety of partners and the complexity of forms of cooperation demonstrate clearly the full complexity and site-specificity of challenges.

Practice has shown that various influences, such as spatial distance, differences in culture and language, inconsistency of processes, as well as intransparency of information and communication can be reasons for a failure in cooperation, at the operative work level alone. Coordination and exchange of information between participants in a distributed product development team is technically difficult and time-consuming, while different locations and time zones further complicate communication.

The term ‘distributed collaborative engineering’ intends to accentuate an approach to the investigation of collaboration and product development in geographically-distributed environments and cross-enterprise contexts, which does not exclude an interdisciplinary approach.

1.1.2. Interdisciplinarity

This current trend in engineering will be presented with the help of an example. When people started making cars over a hundred years ago, the key focus was on the mechanics of the vehicle. Electrics was relegated to the background and limited to a few components such as windscreen wipers or lights (cf. [Iglsböck, 2002]). Mechatronics was first used with respect to the computer control of electric motors by an engineer at Japan's Yaskawa Electric Co. in the late 1960s (cf. [AIDC, 2006] and [Yaskawa, 2006]). This term remains popular in Japan, where it usually refers to a fusion of mechanical parts and a broadly defined electronics. Mechatronics technology has continued to advance rapidly since the coining of this term. These advances are concerned largely with precision, speed, durability, miniaturization, flexibility, safety, power consumption and cost.

The term ‘mechatronics’ today, means the integration of electronics, software and mechanics. Mechatronic systems consist of sensorial, processing and actuating elements like measurement systems, engines, valves, controllers, Asics and microprocessors [Scherber, 1997]. The purpose of this interdisciplinary engineering field is the study of automation devices from an engineering perspective and serves the purpose of controlling advanced hybrid-systems such as production systems, synergy-drives, and everyday equipment such as auto focus cameras, video, washing machines, etc. Mechatronic systems' behaviour is determined by interdependencies between different components. Therefore, an integrated and interdisciplinary engineering approach is necessary. Communication across the traditional boundaries between mechanical, electro-technical and software engineering must take place at an early design-process stage.

Only a multi-disciplinary team of professionals can undertake such a complex project. It requires interaction and communication between the people involved. Additionally, it often relies on the ability of a cross-domain or interdisciplinary team to create a shared understanding of the task, the process, and the respective roles of its members.

It cannot be denied that the current trend in engineering is to interconnect different functions and fields of work during the entire development process: this is called ‘interdisciplinary engineering’ or ‘cross-domain engineering’.

1.2. Problem

It is evident that today's new engineering processes connect more strongly all the various disciplines together, especially in all ranks: integrated at the process level as well as through communicative and informative support. Moreover, globalisation has created a need to collaborate and compete with counterparts located thousands of miles away. When a project is in a cross-enterprise/cross-domain context, unconnected islands of knowledge (see Figure 2-3) need to be merged.

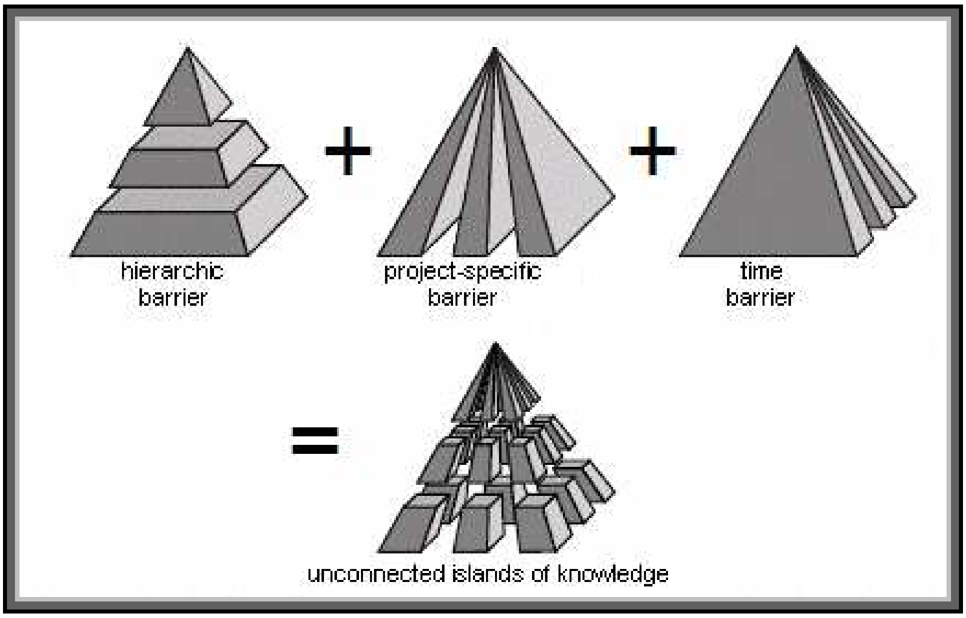


Figure 1-1: Islands of knowledge resulting from the organisational structure (according to [Probst et al, 1997]).

Significantly, the persons involved must learn to cooperate with each other on common tasks. To coordinate their activities communicating and exchanging information is indispensable (cf. [Reichwald, 1999]). Information and communication systems help such projects in becoming a success by supporting exchange between multiple partners. However, it's obvious that every project has a specific character and that a single IT-tool cannot support every engineering activity on its own. Moreover, choosing the most appropriate IT-support or configuring an adequate IT-platform is quite difficult, since technology is always in evolution and defining precise requirements on collaboration software is extremely complex. Consequently, starting a new development project often needs a time-consuming and costly analyse of the project profile in order to define an adequate IT-support. In fact, characterising a new project is often very difficult, since most of the characteristics depend on organisational and human factors and thus show fuzziness. Subsequently, defining the right IT-support frequently requires the intervention of experts which generates new costs and delays an efficient start of the project.

1.3. Goal

This thesis presents a methodology to configure IT-collaboration-platforms appropriate for a given project profile. It proposes a taxonomy system of cross-enterprise and cross-domain engineering factors and a classification of IT-tools with respect to their functionality. It also evaluates the impact of each factor on the usability of these IT-tools and creates rules to determine when one tool is more appropriate than another. In essence, this dissertation proposes to answer the following questions:

- What are the characteristics and limits of the new engineering trends 'cross-enterprise/cross-domain engineering'?
- How can cross-enterprise/cross-domain engineering be classified? What are their principal characteristics and limits? Which factors are to be considered and how are they to be evaluated?
- How can an appropriate IT-collaboration-platform, more exactly, its functionality, regarding the resulting classification be configured?
- How can a correspondence between cross-enterprise/cross-domain factors and IT-tools be established? Which rules must followed?
- Does this method of configuring IT-collaboration-platforms appropriately for specific project configuration have a repercussion on global economical benefits?

We will try to answer these questions with a new systematic approach in decision-making in the IT and especially in the field of collaboration. The methodology uses empirical data collected on our own and the conception of an expert-system based on fuzzy logic, in order to make decision in the selection of IT-tools. The method to be developed also proposes the formalisation of vague human factors, which are very complex to define. It will then be verified by applying it to real development projects.

In conclusion, the contribution of this work is to provide a method helping in the choice of IT-tools for cooperation projects regarding their functional aspects. In contrast to other approaches in this field of research (see section 1.5), the methodology to be developed shall be independent from the processes followed in the project, as well as from the tasks to be executed, and thus, can be applied to each kind of engineering projects, such as software engineering, mechanical engineering, industrial engineering, method engineering, civil engineering, etc. It shall be based on influencing factors of new engineering trends and propose a formalisation of complex human factors with the help of fuzzy logic, as well as configuration-rules for IT-platforms, based on a self- and custom-made empirical study. The proposed methodology is of great significance to reduce "time to decision" processes in the early phase of a development project, especially if the project must start quickly and needs a quick estimate of the global IT-costs.

1.4. Related Work

The available related work can generally be classified by three categories: cross-domain and cross-enterprise engineering, collaborative and virtual engineering, and fuzzy expert-systems. Works on cross-enterprise and cross-domain contexts are diversified and explored by researchers from many disciplines, often relating to collaborative and virtual work as well.

Schleidt, a psychologist, and Eigner, a specialist on product-lifecycle management, work on very similar topics. They try to define personal success factors for cross-enterprise and cross-domain contexts and propose definitions of cross-enterprise, cross-domain and virtual project work (cf. [Eigner et al., 2006]). Kreimeyer, another researcher in the field of product engineering also has a similar approach to our work and tries to develop a methodical support of multi-disciplinary globally-distributed design projects. The methodology provides guidelines and instructions on how distributed teams involving members from several disciplines should be run. However it is more a design-engineering point of view than an IT perspective (cf. [Kreimeyer et al., 2006]).

Approaches to decision-making in the selection of IT-tools are often based on a precise analysis of processes and tasks. In [Zizala, 2006], for example, the author developed a method, which determines the necessary IT-applications supporting specific cooperation processes. The resulting set of applications is then called an IT-Package. However this approach is very specific to the processes under consideration, and neither considers other engineering processes, nor cross-domain or cross-enterprise processes.

Fuzzy-expert-systems often have the same approach as the one used in this work. However, these are principally used in wide-ranging fields, including linear and nonlinear controls, pattern recognition, financial systems, and operations research (cf. [Mizumoto, 1989]) rather than to make strategic decisions regarding IT-solutions. Only few researchers use a fuzzy-expert-system approach to collaborative work. In [Zhang et al., 2006] they use an expert-system which combines cooperative theory and speech-act theory to propose first an organizational relationship graph of role, behaviour and rule of CSCW systems (Computer Supported Cooperative Work) and then a logical predication-based model. Another example is the fuzzy group-preferences analysis method for new-product development presented in [Lo et al., 2006].

1.5. Structure of Content

First, the two notions, cross-enterprise engineering (CEE) and cross-domain engineering (CDE), that are derived from current trends of engineering will be defined by pointing out why working in a cross-enterprise and/or cross-domain engineering context is necessary as well as the context's characteristics, difficulties and limits.

The thesis will then propose a method to characterise engineering projects regarding their distribution and interdisciplinarity. This classification uses influencing factors of CEE and CDE for the engineering process under consideration and details the project profile. We will rely on and take recourse to results of previous analyses to determine these factors and their influence.

For the cross-enterprise part we will work with the characterisation system of Gierhardt and Anderl (cf. [Gierhardt et al., 1999], [Anderl et al., 1999a] and [Anderl et al., 1999b]). On the basis of theoretic considerations as well as experiences acquired by industrial development projects, they have developed a characterisation system (MMS, Merkmalsystem) to classify cross-enterprise projects. Every factor has a parameter value that characterises the degree of distribution which is specific to the project or the process. All in all the system works out 15 influencing factors together with their parameter values.

To characterise the development project on the cross-domain level, we will base our analysis on Hartmann's work (cf. [Hartmann, 1998]). This study on controlling interdisciplinary projects, a theoretic as well as an empirical work, presents recommendations for organising such projects. Although this work depicts influencing factors for interdisciplinary projects, the evaluation system must nevertheless be adapted to our needs. In fact, not all factors can be correlated to industrial-development projects and are only applicable in the academic environment. Subsequently, we will first extract the factors corresponding to industrial development projects and then organise our classification system.

The classification's goal is to bring out the specific characteristics of development projects to configure an appropriate IT-collaboration-platform. We will not design the entire IT-system, but will concentrate our work on the configuration of its functionalities.

Toward this end an analysis of existing IT-tools will be made with the help of previous studies on groupware systems, e.g. the classification schema for CSCW-applications according to [Teufel et al., 1995], or the formalization of CSCW-applications according to [Chang et al., 2001]. Each tool will be described with respect to its functionality. The analysis will be extended to specific tools from collaborative engineering according to [Mills, 1998].

The influencing factors as well as the description of the various IT-tools will give us the possibility to construct rules to configure an appropriate IT-platform. Toward this end we will use fuzzy logic, thus creating a fuzzy-expert-system. Fuzzy logic is derived from the fuzzy-set theory, which deals with reasoning that is approximate, rather than precisely deduced from classical predicate logic. It can be thought of as the application side of fuzzy-set theory dealing with well thought-out real world expert values for a complex problem (cf. [Klir, 1997]).

First, the factors will be defined according to the fuzzy-set theory. Each factor definition will be justified separately, if possible with the help of previous analyses. Since the rules depend on psychological and sociological factors, studies about human preferences, capabilities and tendencies must be taken into consideration. In the absence of previous justifying works, we will use empirical methods to demonstrate our definition of fuzzy sets.

The expert-system will finally use the results of the empirical research as a knowledge-base to construct the rules as well as to define the impact of each factor on the configuration of the IT-collaboration-platform. It will then be implemented in the form of a software tool.

Thereafter, the method will be verified by applying it to two product-engineering projects at BMW and a software engineering project at the TU München. They will first be classified to bring out the project profiles. We will then use the defined methodology to identify the necessary functionalities in order to configure an appropriate IT-support system. Finally, the generated configuration will be implemented in a BMW Group existing IT-platform[1]. The purpose of these case studies is first to evaluate the developed method, then to identify its limitations and finally to point out aspects of consideration for the classification and implementation of the generated IT-platform.

Last, but not least, the concept will be evaluated in a more global manner. One important criterion for the evaluation is the feasibility of the concept to support other engineering projects; another one is the global benefit of such an expert-system and method to configure IT-support systems. In fact, it’s important to note that choosing the right IT-system can have a great impact on the development project in terms of cost reduction, time reduction, and higher quality. For this reason it is very important to define the characteristics and needs of a project very precisely. We will demonstrate that by using this methodology new collaboration projects are set up faster and with low costs. A brief economical study about the benefits made by the implementation of such a method will be presented.

The following diagram (cf. Figure 1-2) summarizes the intended scope of this work.

[1] Virtual Project Space (VPS), a web-application for collaborative engineering integrated in the Partner Portal of the BMW Group (see section 4.7.1.1).

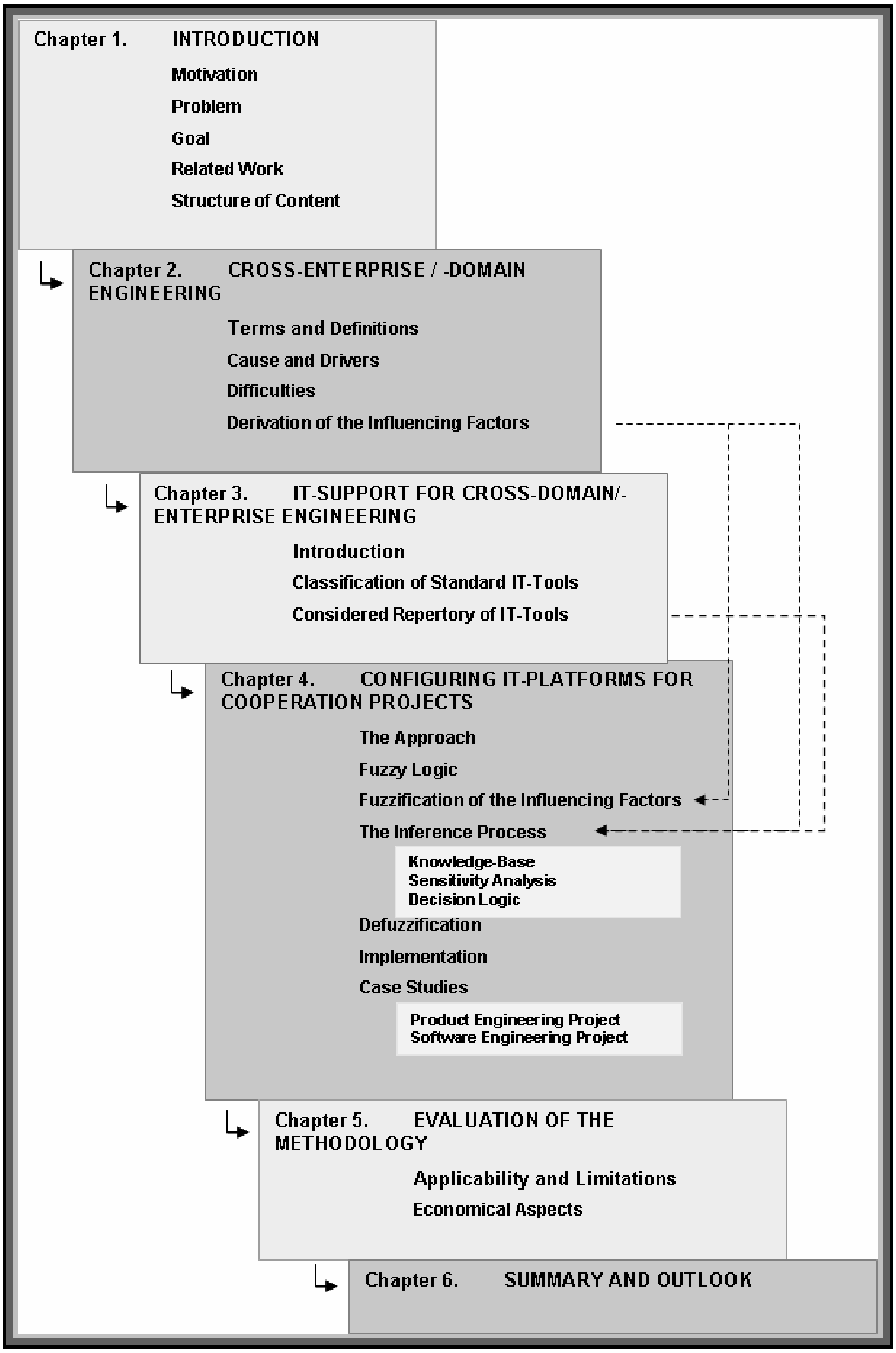

Chapter 1. INTRODUCTION
Motivation
Problem
Goal
Related Work
Structure of Content
Chapter 2. CROSS-ENTERPRISE / -DOMAIN ENGINEERING
Terms and Definitions
Cause and Drivers
Difficulties
Derivation of the Influencing Factors
Chapter 3. IT-SUPPORT FOR CROSS-DOMAIN/-ENTERPRISE ENGINEERING
Introduction
Classification of Standard IT-Tools
Considered Repertory of IT-Tools
Chapter 4. CONFIGURING IT-PLATFORMS FOR COOPERATION PROJECTS
The Approach
Fuzzy Logic
Fuzzification of the Influencing Factors
The Inference Process
Knowledge-Base
Sensitivity Analysis
Decision Logic
Defuzzification
Implementation
Case Studies
Product Engineering Project
Software Engineering Project
Chapter 5. EVALUATION OF THE METHODOLOGY
Applicability and Limitations
Economical Aspects
Chapter 6. SUMMARY AND OUTLOOK

Figure 1-2: Structure of the thesis.

2. CROSS-ENTERPRISE / -DOMAIN ENGINEERING

The current trends described in 1.1 points out two important aspects of engineering prevalent today: first, the distribution aspect, i.e. the development in geographically distributed environments with cross-enterprise engineering (CEE) and, secondly, the interdisciplinarity or cross-domain engineering (CDE), which alludes to the interaction of specialists from different domains.

2.1. Terms and Definitions

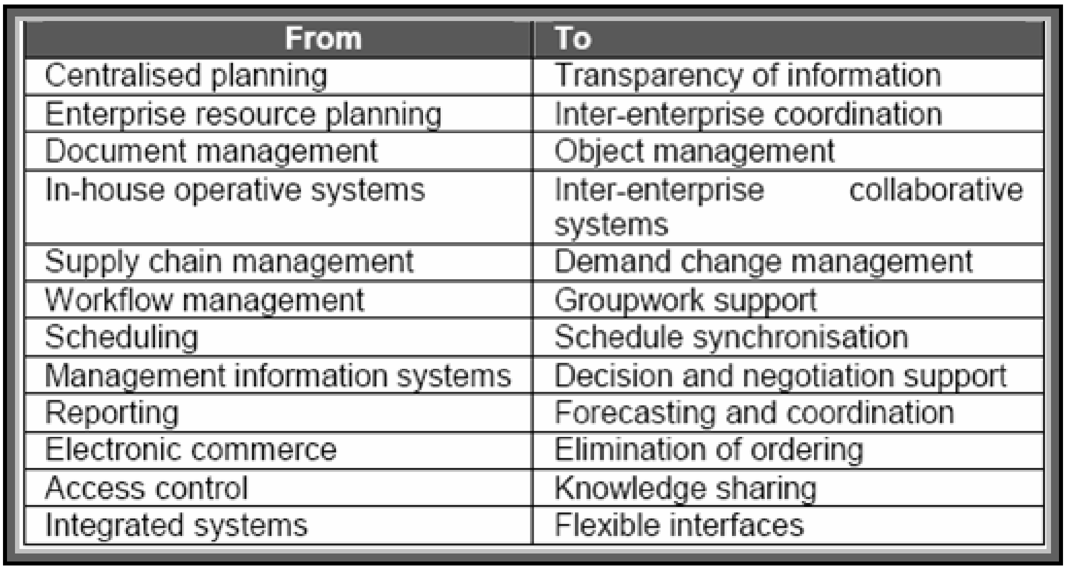
The terms ‘cross-enterprise’ and ‘cross-domain’ may seem confusing. It is in fact evident that they are related and that each of them influences the engineering process. Subsequently there is a need to define these terms and then to establish their relationship, pointing out characteristics, needs and critical points of interdisciplinarity and distribution.

2.1.1. Cross-Enterprise Engineering (CEE)

In the context of the globalisation the term ‘cross-enterprise engineering’ means all engineering activities along a product’s or software’s lifecycle that cross the enterprise’s boundaries.

A cross-enterprise collaboration is frequently distributed and accordingly virtual, namely the fact that the workers do not sit in the same room, but rather at different locations either within the same enterprise or beyond its geographical limits. This distribution varies from the possibility of several groups at the same place to purely virtual teams, where everyone is dispersed. Communication takes place predominantly with the help of electronic media, like E-Mails, Internet, Groupware, etc., by phone or video-conferences and, but more rarely, as a face-to-face event. The form of collaboration depends on the locality of each participant and whether they are in the same time zone or not. Other factors must also be taken into account. First, every project has its own defined cross-enterprise organisation. Secondly, it is evident, that such projects make the relations and interactions more complex. The complexity of a face-to-face meeting between two persons of different cultures alone shows clearly the insufficiency of relying on information technology or project management tools only.

Cross-enterprise engineering is often also named engineering collaboration, which is strictly speaking the same type of corporate-spanning collaboration or inter-enterprise engineering. According to [Kazi et al., 2001], inter-enterprise collaboration is gaining momentum with the emergence of new information and communications technologies to support exchange and collaborative work amongst distinct geographically dispersed entities. These new project configurations are changing the way organisations function and collaborate with each other. New trends are emerging and priorities are merging like shown in Table 2-1.



From	To
Centralised planning	Transparency of information
Enterprise resource planning	Inter-enterprise coordination
Document management	Object management
In-house operative systems	Inter-enterprise collaborative systems
Supply chain management	Demand change management
Workflow management	Groupwork support
Scheduling	Schedule synchronisation
Management information systems	Decision and negotiation support
Reporting	Forecasting and coordination
Electronic commerce	Elimination of ordering
Access control	Knowledge sharing
Integrated systems	Flexible interfaces

Table 2-1: Changing trends and priorities for inter-enterprise collaboration (according to [Kazi et al., 2001]).

Consequently, the provisioning of information and communications technologies that support dynamic, geographically and organisationally dispersed project teams is a key area of both research and development.

The cross-enterprise context has, of course, an impact on engineering processes. In fact, despite the increased complexity of network processes in comparison to internal processes, those involved have to adapt to constantly occurring changes in a fast and flexible way (cf. [Mertens, 1995]).

2.1.2. Cross-Domain Engineering (CDE)

Cross-domain engineering is a new understanding of the processes involved in design and engineering and the way in which they interact (cf. [Encarnação, 2005]).

[Eigner et al., 2006] define CDE as the engineering activities along the product's or software's lifecycle, which contribute toward the interaction of specialists of different domains. The term 'cross-domain' in industry means 'interdisciplinarity' for academicians and researchers.

Interdisciplinarity does not have a standard basic understanding. In fact, a lot of other terms like 'multidisciplinary', 'trans-disciplinary', or 'cross-disciplinary' also exist. Nevertheless, there is a general accordance that interdisciplinarity means more than just a juxtaposition of different theories to a common theme, but rather the encouragement of a cross-discipline dialog between representatives of each discipline and, accordingly, the creation of synergies (cf. [Hübenthal, 1991],[Heckhausen, 1987]).

Operating across disciplines, domains or fields of knowledge, alternatively also towards their own limits, is currently a new task that the developers promote, as well as altering their methods of work. According to [Garbay, 2006] and [Tchounikine, 2002], practicing interdisciplinarity means taking advantage of this plurality; it means guaranteeing the diffusion of ideas and mobility of concepts. Concepts produced this way are characterized by their operative character. The object's finality, its internal characteristics and the environment in which it's placed are of interest.

Interdisciplinary or cross-domain cooperation thus implies anchoring the necessary efforts of mutual explication within the dynamic caused by the interrelationships of the elements, as opposed to considering them in isolation, as disembodied. The information sciences have an essential role to play here which contributes to the conception and creation of artificial systems that allow users to represent themselves, to understand and intervene upon a certain reality (cf. [Garbay, 2006] [Dubois et al., 2002]). Especially in the CSCW field, interdisciplinarity has its own theory. [Bannon, 1992] describes a more radical approach to the concept of interdisciplinary theory that has been the attempt to 'wed' different disciplines together – for instance, by constructing common dictionaries of terms and concepts, which the different disciplines are supposed to utilize, or making mappings across conceptual frameworks, thus attempting to ensure some 'shared understanding' among researchers.

The only question now is whether the information-processing sciences are able to join this search for interdisciplinarity.

2.1.3. Cross-Cultural Aspects

The notions of cross-domain and cross-enterprise engineering point out another aspect of these activities namely, the cross-cultural one. In fact, on one hand, the internalisation of corporate activity and the development toward the enterprise's limits force the work between different organisations and eventually from different countries. This observation introduces two notions of culture: the organisation's culture and the country's culture.

On the other hand, CDE adduces a third type of culture specific to each discipline.

The definition of culture proposed by [Hoult, 1969] suggests a common way of understanding culture seeing it as consisting of four elements that are 'passed on from generation to generation by learning alone': values, norms, institutions, and artefacts.

Due to the cross-enterprise or cross-domain character, the project's participants may have their own country's, enterprise's and respectively discipline's culture: other language, priorities, approach to the goal, attitude, exposure to complexity, practices of management, as well as processes, tools methods, and so on.

In such engineering projects the acceptance and overcoming differences in culture between the different partners is a factor of success. In fact, the culture defines the frame for everyone's behaviour and team's building. For sociologists this is a very complex topic, which requires a clarification of the subject.

According to [Griffin, 1999] the following aspects describe a culture:

- the culture reflects learned behaviours,
- the culture is adapted to given boundary conditions,
- the culture is shared and defines the integration of oneself within the society.

The culture of a single nation is characterized by [Deresky, 2000] through a set of correlated variables. That way the work habits of a group or individual can be described through motivation and engagement as well as through work ethic and efficiency. These characteristics are based on the type of attitude toward work, materials, individuals, changes or the perception of time (e.g. punctuality).

This attitude is influenced by norms, values and convictions. National influencing factors, such as the law or the economical system, determine group and individual behaviour as well as socio-cultural aspects such as religion, education or language. Subsequently one can refer to a discipline's culture, since each discipline has in fact its own education and dependency on norms and values (cf. [Gierhardt, 2001]).

The cultural context must be taken into consideration without eliminating the differences, but rather by using them to increase creativity. It is difficult to be an expert in another culture; nevertheless it is possible to develop a certain flexibility toward taking the other one's place, psychologically or intellectually.

The terms 'cross-domain' and 'cross-enterprise' are not self-exclusive. A project can have cross-enterprise aspects as well as cross-domain ones, which generally speaking imply a cross-cultural character.

2.2. Cause and Drivers of these Trends

The success of an enterprise is basically characterized by the maximisation of benefits. According to [Danner, 1996] the following factors are fundamental success factors of an enterprise as well as objective-setters to maximise the performance:

- Quality, as the effective and efficient compliance of explicit and implicit wishes of the client,
- Costs, whose limit is set by the concurrence on the market, and
- Time-To-Market.

The aggressive international competition implies the increase of a product's complexity as well as the shortening of a product's lifecycle on the market, which requires a constant optimisation of the product's quality, a reduction of the development costs, and the shortening of engineering time to keep it on the market (cf. [Danner, 1996]).

This complexity of current products requires more and more accurate knowledge. Therefore, organisations have two possibilities: to increase their capacity in order to develop their internal know-how or to hold this know-how at external sources, and so work in a cross-enterprise context.

Another reason mentioned by [Danner, 1996] is the reduction of engineering cycles (see Figure 2-1). For instance, one way to reduce the time of engineering projects is the creation of cross-enterprise task-forces to solve rapidly acute problems.

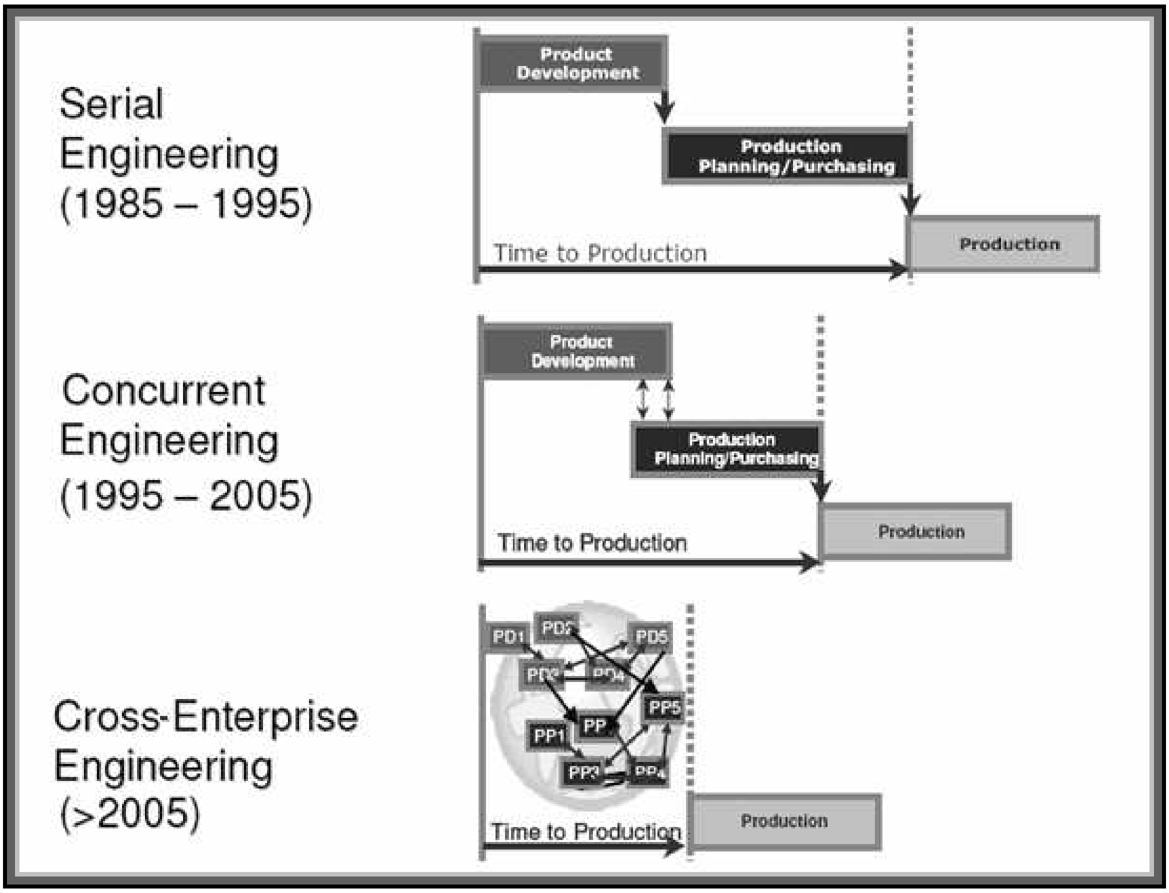

Serial Engineering (1985 – 1995)
Product Development
Production Planning/Purchasing
Production
Time to Production
Concurrent Engineering (1995 – 2005)
Product Development
Production Planning/Purchasing
Production
Time to Production
Cross-Enterprise Engineering (>2005)
PD1
PD2
PD5
PD3
PD4
PP5
PP1
PP
PP3
PP4
Production
Time to Production

Figure 2-1: Evolution of the engineering process (according to [Eigner et al., 2007]).

Moreover, the Economy of Scope is a situation that arises when the cost of performing multiple business functions simultaneously proves to be more efficient than performing each business function independently (cf. [Investor, 2006]), and through it forms a network which uses synergies of the different businesses. It is therefore possible to get costs' advantages through the consolidation of research and development of various enterprises.

Through the cross-enterprise context they have the opportunity to integrate different resources into the enterprise's processes and so to improve the turnover factor of business success. These directions of optimisation are also defined by the criteria costs, time and quality.

It is also of the utmost importance to consider the aspects of flexibility and innovation (cf. [Picot et al., 1998] und [Ehrlenspiel, 1995]). The networking and integration of know-how is a great opportunity to improve the enterprise's acts regarding innovation and the product's quality. For instance, the Centre of Technology of the BMW Group and DaimlerChrysler in Silicon Valley in California has the goal to interconnect know-how of high-tech enterprises with the know-how of the two German Original Equipment Manufacturers (OEM), thus integrating it in both enterprises.

In the same way the following observation explains, for example, the appearance of CDE contexts: due to the increasing penetration of technology in all sectors of society and life, the questions of acceptability, ergonomics and use are given central consideration. The inclusion of concerns over use in the process of conception and development becomes crucial [Garbay, 2006]. This observation introduces a new way of engineering: function engineering, an excellent example of an interdisciplinary process, which is embedded in a complex organisational environment and in which the functional part of a product is highlighted. It covers, on the one hand, the development of functions represented through a software code, which makes the control of an electronic system possible and, on the other hand, the application of the control unit, i.e. its integration into the entire system. Therefore, function and hardware development are tightly connected. Moreover, function engineering also has to be integrated into a complex organisation matrix which often requires collaboration of specialists from various domains. In fact, functional development is a central process of a complex product engineering process in which different disciplines must participate.

Another example is the project FARO at BMW. This CDE project connects all engineering departments at BMW, such as carriage, chassis, engine, design, traffic engineering, strategy, etc., to design visionary scenarios and analyse the direction for the driving dynamics of the future. Although this project is headed by the chassis engineering department, it must be remembered that the chassis does not act isolated in a car. It depends on other factors and components and it influences them as well which is why they are working in a cross-domain context. For that reason FARO initiated a project to be headed by the car architecture and integration department and to involve all the engineering departments.

Cross-domain contexts have been spread out widely, especially in the space of R&D fields. For instance, at the 7th EC-GI&GIS Workshop (European Commission Activities related to Geographic Information and Geographic Information Systems) organised by the Space Application Institutes, it has been declared that notwithstanding the development of sector specific standards, it's also widely recognised that users need to be able to discover resources across different sectoral and disciplinary domains [EC-GI&GIS, 2001]. Another obvious reason is that cross-domain working is a stepping stone to creativity like [Johannsson, 2004] stated: "When you step into an intersection of fields, disciplines, or cultures, you can combine existing concepts into a large number of extraordinary new ideas."

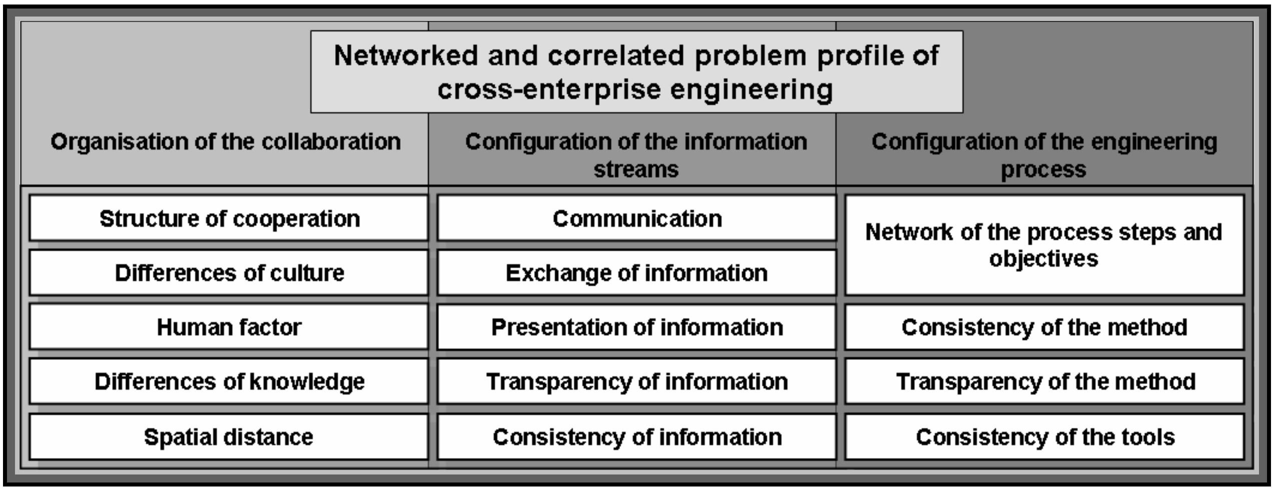
2.3. Difficulties of these Engineering Processes

The CEE defined in §2.1.1 is perceived as an additional difficulty to the normal project work, due to the job's conditions. In the majority of cases the participants are also involved in various projects. Generally this circumstance makes up the relations more complex for everyone.

This work context meets challenges in the following domains [Eigner et al., 2006]:

- Coordination, planning and organisation
- Communication (linguistic or cultural difficulties)
- Complexity due to the new work context (e.g. increasing number of interfaces, complex decision process, etc.)
- Management of distributed teams
- Exposure to new media and IT-systems
- Performance and pressure of time
- Teamwork
- Overcoming cultural differences
- Transparency of knowledge and information

Concrete problems can be derived from these challenges. Figure 2-2 describes the problem profile of CEE. In most cases these difficulties are not technical ones, but so-called 'soft' problems, like insufficient arrangements, inconsequent procedure or lack of sensibility when dealing with other cultures.



Networked and correlated problem profile of cross-enterprise engineering

Organisation of the collaboration	Configuration of the information streams	Configuration of the engineering process
Structure of cooperation	Communication	Network of the process steps and objectives
Differences of culture	Exchange of information	
Human factor	Presentation of information	Consistency of the method
Differences of knowledge	Transparency of information	Transparency of the method
Spatial distance	Consistency of information	Consistency of the tools

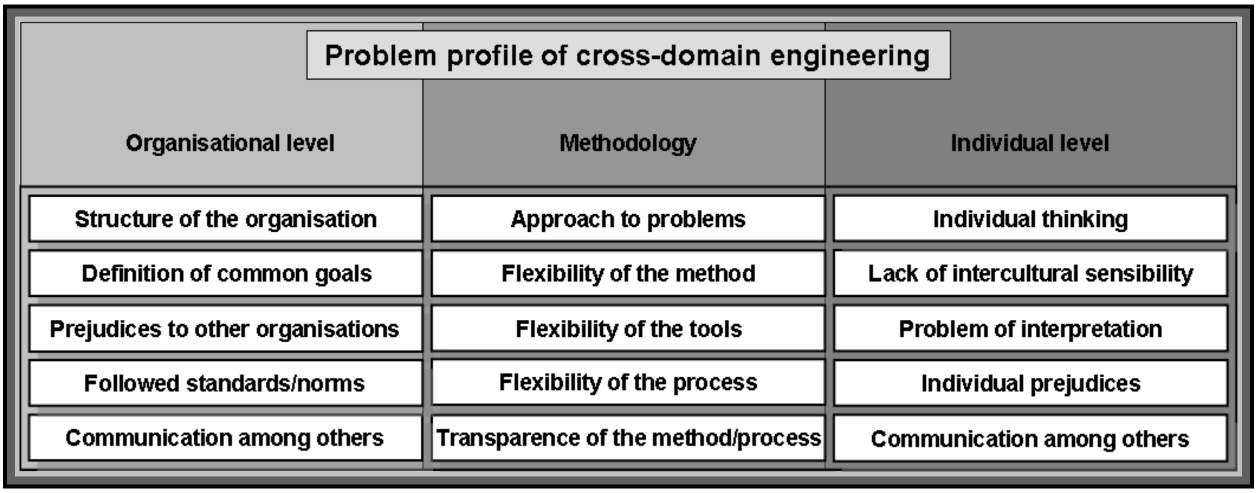
Figure 2-2: Problem profile of the cross-enterprise context (in accordance with [Gierhardt, 2001]).

Numerous questions arise from this problem profile, such as how cultural differences are to be overcome or how the project is to be organised (e.g. which team structure, responsibilities, etc.). In this case organisation units in which an agreement process already exists are recommended, among others. The overcoming of cultural differences, which is represented through, e.g., various languages and educations or the special manners and customs, also presents a considerable problem (see §2.1.3). Differences of knowledge could also be obstacles to cooperative work and CEE, particularly with regard to the definition of the task, but also to the correlation of process, product and organisation by the participants. The human factor describes a personal approach to teamwork in a distributed environment, including the social competences of the participants and all factors that influence one's attitude. Furthermore, to collaborate over the time and spatial distances is a big challenge, which influences all of the above-mentioned problems. Some other points are put into question, such as how to establish the contact between the participants and how to maintain it so that collaboration works effectively.

The realisation of communication processes is the base of exchanging information in a CEE context. The structure and coordination of this exchange as well as the consistency and transparency of information is also of the utmost importance. Transparency of information refers to the possibility of accessing the project's information and comprehending it, whereas consistency of information refers to its validity and accuracy. Another difficulty is the appropriate representation and presentation of information in cross-enterprise contexts, i.e. the preparation, structure and report for the partner.

An optimal networking of the phases of the process is also essential to realise cooperation in a distributed context. Therefore, it is worth noting that an appropriate distribution of the process phases is obligatory as well as the configuration of the procedures at the junctions between the partners. In addition, it must be remembered that cross-enterprise projects require a collective determination of the goal and, as the case may be, a cross-linking of every partner's goal. Guaranteeing the transparency and consistency of methods and tools, respectively, suggests the possible problems of continuity and compatibility of the engineering methods and tools adopted by the partner. Thus, the adaptation and reorganisation of methods and tools for the elements of CEE are also a basic challenge.

An engineering project that extends across disciplines also meets obstacles. Working across the domains, fields of activity and knowledge is a new exercise which stimulates the participants, but also changes their manner of work. Every discipline, or in some cases community, has its own vocabularies, presuppositions, priorities, criteria, methods, tools, standards and references. Even considering the various benefits of CDE, a lot of obstacles prevent such projects. Apart from hardly knowing each other, not to mention the goals, activities, topics and logic of the other actor, obstacles to cross-domain cooperation may be identified, in general, on three different levels (see Figure 2-3).



Problem profile of cross-domain engineering

Organisational level	Methodology	Individual level
Structure of the organisation	Approach to problems	Individual thinking
Definition of common goals	Flexibility of the method	Lack of intercultural sensibility
Prejudices to other organisations	Flexibility of the tools	Problem of interpretation
Followed standards/norms	Flexibility of the process	Individual prejudices
Communication among others	Transparence of the method/process	Communication among others

Figure 2-3: Problem profile of the cross-domain context (in accordance with [Segebart et al., 2000] and [Hartmann, 1998]).

The organisational level describes the problems relative to the organisation, for example, its structure and standards, which might also pose an obstacle to a successful cooperation, if they are not integrated into an interdisciplinary context. Basically, an organisation is a group of people intentionally organised to accomplish an overall, common goal or set of goals (cf. [McNamara, 1999]). In our case, an organisation can be a group of specialists from the same field with the same educational background, but also a group of various specialists working on the same product. Coordinating this common goal with the other interacting organisations is also a big challenge. Moreover, such an organisation may have its own technical terms and vocabulary, which makes the communication more difficult. In general, different disciplines have different sub-cultures, and the difference becomes worse, not attenuated, by the existence of superficial similarities, for instance, when identical words with quite different meanings are used. [Sperber, 2003] reports about his experience in cross-domain projects, observing that it is fascinating and somewhat disheartening to watch how week after week, year after year, the same agreements across and sometimes within domains and disciplines are expressed in almost the same terms, as if disciplinary and theoretical affiliations could never be overcome.

Exposure to different domains requires openness, rather than thinking according to models and schemas from one's own discipline with bias or prejudice towards what is new and unfamiliar. Prejudices against other disciplines and domains are very frequent. When applied to social groups, prejudice generally refers to existing biases towards the members of such groups, often based on social or educational stereotypes; and at its most extreme, results in groups being denied benefits and rights unjustly or, conversely, unfairly showing unwarranted favour towards others. Consequently, a lot of problems arise. For instance, more often than not, the expectations of the project's participants are not so much a desire to learn from the other discipline, but rather that people from the other discipline can and should learn from them. (cf. [Sperber, 2003]).

The fact that every domain also has its own approach to problems as well as its own methods and processes is of the utmost importance. It is worth noting that it is difficult to lead such a project if everyone uses his/her own methods and tools. The adaptability, transparency and flexibility of the used methods are a condition for CDE, as well as the possibility to integrate the tools and processes of every domain. In fact, in the absence of a systematic approach to cross-domain projects, there is a need for integrated and flexible methodologies and processes.

On the individual level, an interdisciplinary sensitivity is assumed and expected, although most of the participants have neither training nor experience in cross-domain cooperation. Due to a lack of intercultural sensibility, they meet with obstacles similar to the ones met at the organisational level: the problem of interpretation, comprehension and communication, not to mention prejudices toward others. The specialist or individual has his/her own vocabulary, not only depending on his organisation, but also on his background and knowledge. If he depends too much on a community, group or organisation (e.g. a scientific community, his organisation, a development team, etc.), he will probably lack the required flexibility to be part of such a cross-domain project and even develop prejudices that are the residues of his personal history or his affiliation to a group.

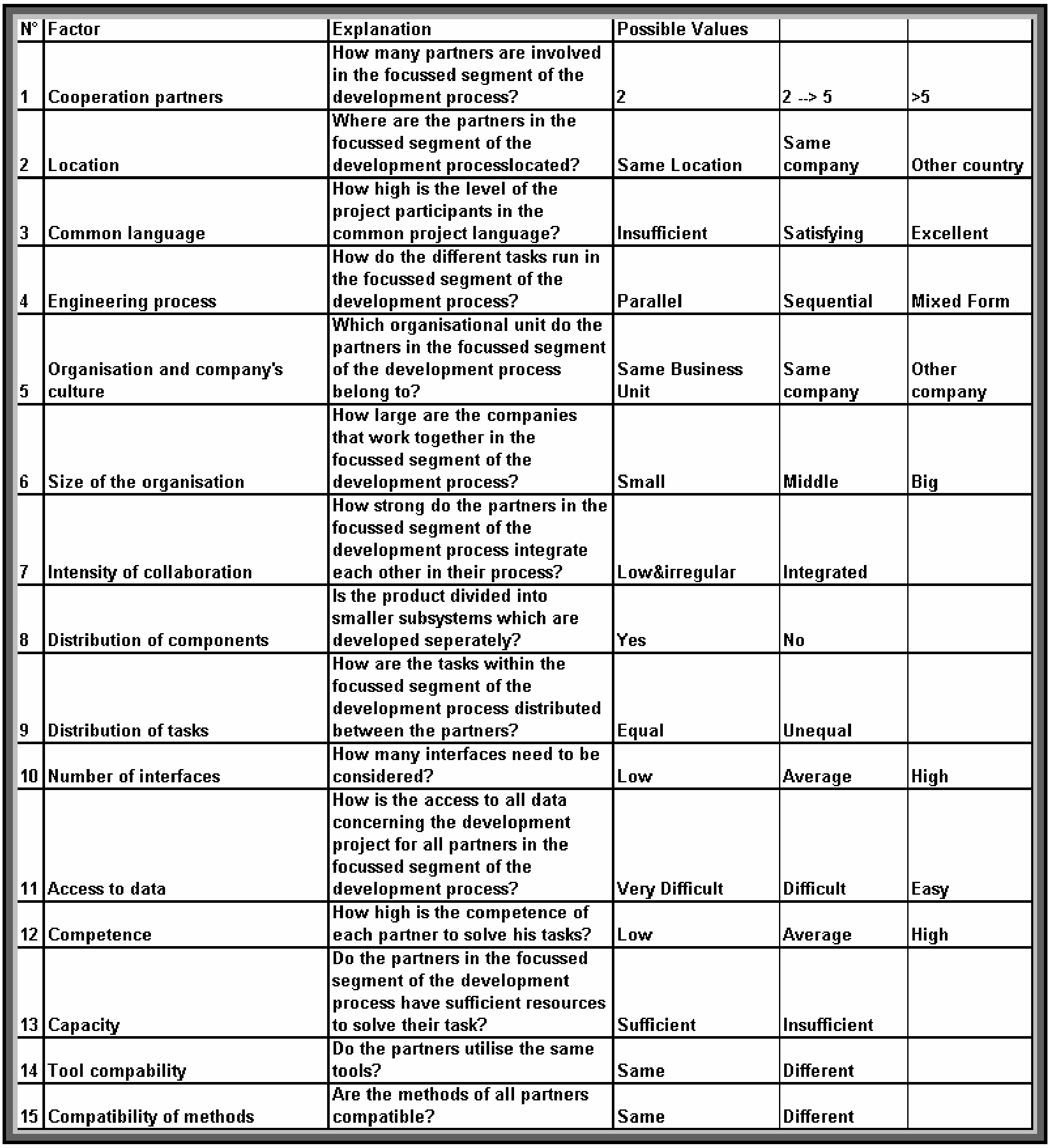
2.4. Derivation of the Influencing Factors

Paragraph 2.1 defines the terms ‘cross-domain’ and ‘cross-enterprise’ as well as their typical characteristics. Cross-enterprise activities, for example, must be differentiated into collaboration within the enterprise (e.g. spatial distance between persons working together) and collaboration across the enterprise limitations. Many factors must be taken into account, such as spatial and cultural distance, quantity of partners, common processes, security aspects, access mechanisms, and so on. All of these factors influence the project’s workflow with more or less impact. For instance, the factor spatial distance: Allen has demonstrated in 1984 that the frequency of communication decreases drastically, when the distance between two persons reaches approximately 10 meters [Allen, 1984].

Similarly the complexity of CDE processes is also influenced by many factors. In fact, it could be a project concerning a team constituted of specialists of different fields as well as teams from different domains working together with a common method. Interdisciplinarity has a variety of aspects, and it’s evident that all these factors influence the progress of an engineering project.

To date no thorough study has been made to determine these factors and their influence. Hence, we use the results of previous analyses.

For the cross-enterprise part we work with the characterization system of [Gierhardt, 2001], [Gierhardt et al. 1999], [Anderl et al. 1999a] and [Anderl et al. 1999b]. On the basis of theoretic reflections as well as experiences acquired by industrial development projects, they have developed a Taxonomy System (MMS, Merkmalsystem), which points out typical problems of CEE and offers possible solutions for these problems. One of the modules of the system is the matrix of characteristics, which contains typical characteristics of distributed engineering processes with their possible conditions. A classification profile of a distributed engineering process consists of the combination of the condition of each single characteristic. In order to give an entire profile, they have elaborated 15 characteristics which are essential for distributed development processes. The 15 characteristics are listed in Table 2-2.



N°	Factor	Explanation	Possible Values		
1	Cooperation partners	How many partners are involved in the focussed segment of the development process?	2	2 --> 5	>5
2	Location	Where are the partners in the focussed segment of the development processlocated?	Same Location	Same company	Other country
3	Common language	How high is the level of the project participants in the common project language?	Insufficient	Satisfying	Excellent
4	Engineering process	How do the different tasks run in the focussed segment of the development process?	Parallel	Sequential	Mixed Form
5	Organisation and company's culture	Which organisational unit do the partners in the focussed segment of the development process belong to?	Same Business Unit	Same company	Other company
6	Size of the organisation	How large are the companies that work together in the focussed segment of the development process?	Small	Middle	Big
7	Intensity of collaboration	How strong do the partners in the focussed segment of the development process integrate each other in their process?	Low&irregular	Integrated	
8	Distribution of components	Is the product divided into smaller subsystems which are developed seperately?	Yes	No	
9	Distribution of tasks	How are the tasks within the focussed segment of the development process distributed between the partners?	Equal	Unequal	
10	Number of interfaces	How many interfaces need to be considered?	Low	Average	High
11	Access to data	How is the access to all data concerning the development project for all partners in the focussed segment of the development process?	Very Difficult	Difficult	Easy
12	Competence	How high is the competence of each partner to solve his tasks?	Low	Average	High
13	Capacity	Do the partners in the focussed segment of the development process have sufficient resources to solve their task?	Sufficient	Insufficient	
14	Tool compability	Do the partners utilise the same tools?	Same	Different	
15	Compatibility of methods	Are the methods of all partners compatible?	Same	Different	

Table 2-2: Cross-enterprise characteristics (from [Gierhardt et al. 1999], [Anderl et al. 1999a] and [Anderl et al. 1999b]).

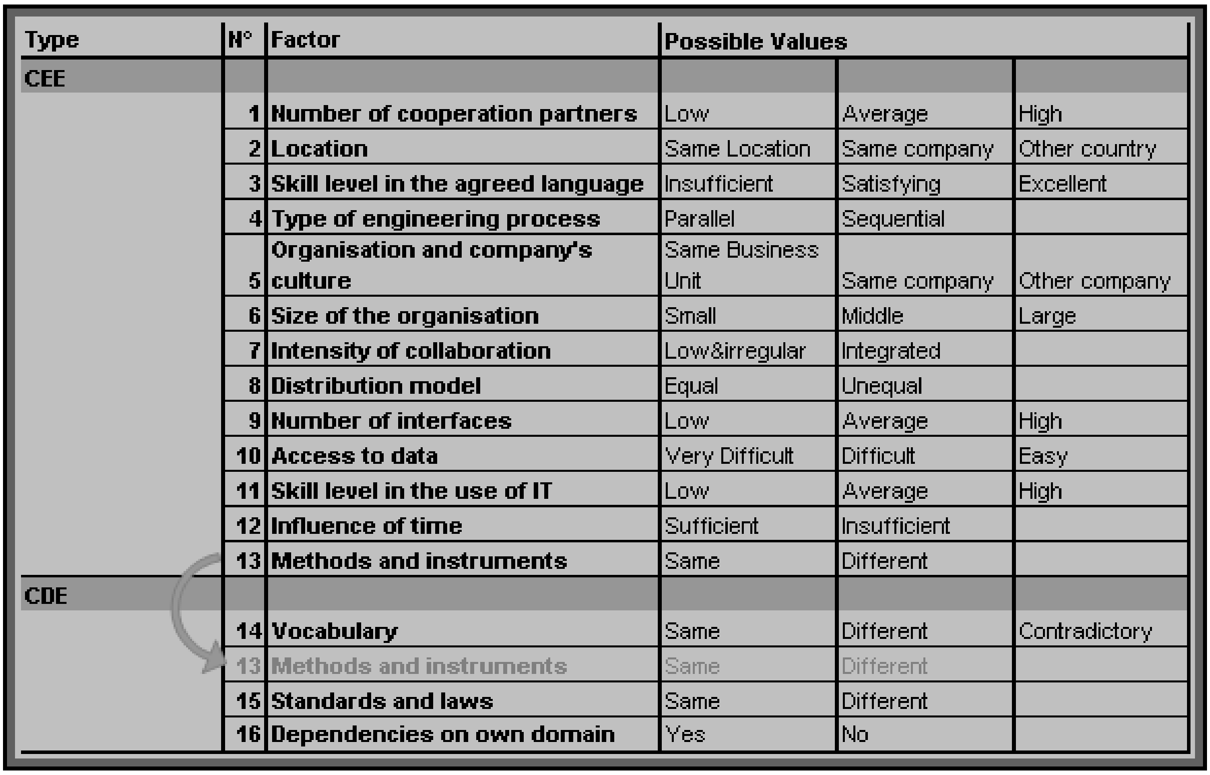
To characterize the development project on the cross-domain level we base our analysis on Hartmann's work [Hartmann, 1998]. This study about controlling interdisciplinary projects, a theoretic work, as well as an empirical one, presents recommendations to organise such projects. This work shows influencing factors for interdisciplinary projects with their impact on three levels: personal characteristics, factors specific to the project and finally, those specific to the disciplines involved. The evaluation system developed by Hartmann must nevertheless be adapted to our needs. In fact, not all factors can be adapted to industrial development projects and are of more concern to academicians. We will subsequently first extract the factors corresponding to industrial development projects and then organize our classification system. [Hartmann, 1998] proposes 11 influencing factors resulting from the interaction of different domains:

- Own terms
- Own paradigm
- Same terms have different meanings
- Own methods and instruments
- Own norms and laws
- Problems of comprehension
- Prejudices to other disciplines
- Relationship of dependence to the own discipline
- Attitude to other disciplines
- Different positions to science
- Talk to same discipline most important

[Hartmann, 1998] extracts then the 4 most relevant factors of interdisciplinary projects which she designates, namely 'Own terms', 'Own methods and instruments', 'Own norms and laws', and 'Own paradigm'.

In the thesis we use a taxonomy of 16 factors, combining the factors of [Gierhardt et al. 1999], [Anderl et al. 1999a], [Anderl et al. 1999b] and [Hartmann, 1998].

The factor 'Own methods and instruments' is closely related to the factors 'Tools compatibility' and 'Compatibility of methods' from Table 2-2, which is why we have paired them. Similarly, [Gaul, 2001] couples the factors 'Distribution of the tasks' and 'Distribution of the components', making one, since they are actually dependent on each other. Some factors have been renamed, so that they match our scope better: configuring IT-platforms for CEE and CDE projects or facilitating their comprehension. Every factor will be described blow-by-blow in the section 4.3. The resulting taxonomy is shown in Table 2-3



Type	N°	Factor	Possible Values		
CEE					
	1	**Number of cooperation partners**	Low	Average	High
	2	**Location**	Same Location	Same company	Other country
	3	**Skill level in the agreed language**	Insufficient	Satisfying	Excellent
	4	**Type of engineering process**	Parallel	Sequential	
	5	**Organisation and company's culture**	Same Business Unit	Same company	Other company
	6	**Size of the organisation**	Small	Middle	Large
	7	**Intensity of collaboration**	Low&irregular	Integrated	
	8	**Distribution model**	Equal	Unequal	
	9	**Number of interfaces**	Low	Average	High
	10	**Access to data**	Very Difficult	Difficult	Easy
	11	**Skill level in the use of IT**	Low	Average	High
	12	**Influence of time**	Sufficient	Insufficient	
	13	**Methods and instruments**	Same	Different	
CDE					
	14	**Vocabulary**	Same	Different	Contradictory
	13	**Methods and instruments**	Same	Different	
	15	**Standards and laws**	Same	Different	
	16	**Dependencies on own domain**	Yes	No	

Table 2-3: Influencing factors of CEE and CDE.

The factors and their possible values provide a basis to classify engineering projects on the cross-enterprise and cross-domain level and then to configure an appropriate IT-platform. This question can only be solved by classifying IT-tools, so that they can correspond to the observed engineering projects. The next chapter defines the categories of IT-tools in order to develop a classifying schema for standard tools as well as for project-specific tools.

3. IT-SUPPORT FOR CROSS-DOMAIN/-ENTERPRISE ENGINEERING

Some of the challenges of cross-enterprise and cross-domain contexts can be solved by or with the help of sophisticated IT-tools and platforms. The main focus of IT-systems for CDE and CDD projects lies on topics like communication, transfer of information, team coordination as well as special cases of the engineering process (cf. [Lassenius et al., 2003]).

3.1. Introduction

The goal is to provide tools and methods so that a geographically distributed team can collaborate as easily as co-located teams (cf. [Tang et al., 1994]). Collaboration can happen synchronously when all participants view information and/or meet at the same time or asynchronously when participants view information and provide feedback at different points in time.

The most common way to support standard processes is to use Groupware Systems. [Ellis et al., 1991] define groupware as computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment. From the first investigations into CSCW in 1984 until now, researchers have achieved a great deal (e.g. [Chang et al., 2001], [Cortes et al., 1996], [Li et al., 1998], [Arisha et al., 1999], [Astley et al., 1998], [McComb et al., 1999], [Persson et al., 2005]). In addition to the standard processes and groupware, collaborative engineering aims to provide concepts, technologies and solutions for development in dispersed engineering teams and thus presents solutions more specific to engineering disciplines.

3.1.1. Computer Supported Cooperative Work (CSCW)

The term ‘CSCW’ was first coined by Irene Greif and Paul M. Cashman in 1984 at a workshop attended by individuals interested in using technology to support people in their work (cf. [Grudin, 1994]). According to [Carstensen et al., 2002], CSCW addresses how collaborative activities and their coordination can be supported by means of computer systems.

Many researchers consider CSCW and groupware to be synonyms. However, while groupware refers to real computer-based systems, CSCW focuses on the study of tools and techniques of groupware as well as their psychological, social, and organizational effects. Thus, CSCW is an interdisciplinary research field in informatics, sociology, psychology, management, anthropology, ethnography, economics, and so on, which is concerned with information and communication technologies supporting team work (cf. [Greenberg, 1991], [Bornschein et al., 1995], and [Hasenkamp et al., 1994]). The definition of [Wilson, 1991] expresses the difference between these two concepts: CSCW is a generic term, which combines the understanding of the way people work in groups with the enabling technologies of computer networking, and associated hardware, software, services and techniques.

The practical realisation of CSCW as research field is defined as groupware and consists of the information and communication technologies supporting teamwork. Influencing factors of groupware are humans, tasks, organisation and technology (cf. [Bornschein et al., 1995]). According to [Greenberg, 1991], ‘groupware’ is software that supports and augments group work. It is a technically-oriented label meant to differentiate ‘group-oriented’ products explicitly designed to assist groups of people working together from ‘single-use’ products that aid people in pursuing their isolated tasks only.

Groupware support principally standard information and communication processes with tools such as E-Mails or Online-Conferencing and will be described in more detail in §3.2.

3.1.2. Collaborative Engineering

According to [Mills, 1998], collaborative engineering is the application of team-collaboration practices to an organisation’s product development endeavours. Also known as collaborative product development, it builds upon the nature of cross-domain product development teams. In essence, it is the union merger combining of concurrent engineering to the concept of highly effective and well-supported team collaboration, including not only the act of collaboration itself, but also the infrastructures and environments that enable and nurture it. Collaborative Engineering creates the technological conditions for collaborations in networks across companies, and it provides the necessary processes and applications (cf. [Mills, 1998]).

Collaborative engineering can also be described as the employment of information technology to establish virtually collocated workgroups, when it was impossible to establish a real physical workgroup. Collaborative engineering, however, is more than just information sharing. It’s a cooperative exchange of resources among a team focused on an engineering project, which may have the same creative purpose and ‘work cooperatively’ to exchange information, ideas, or useful resources to create a shared understanding of it.

Even when teams are not dispersed over large distances, individuals have been using tools to communicate and collaborate with one another for many years already. The types and amounts of systems and data being interchanged within teams as well as the need for collaboration are ever increasing. In fact, collaborative engineering environments and also distributed environments support a great deal of industrial activities. It is beneficial to examine which type of groups collaborative engineering environments can support (cf. [Mills, 1998]):

Product Planning

Product planning is in charge of all initial market research and feasibility analyses and often works in cooperation with designers to produce models of new products. They use tools to collect and analyze voice-of-the-customer data. The most powerful and widely-used tool for this type of analysis is QFD (Quality Function Deployment). They may produce data such as QFD matrices, feasibility results, human resources requirements, scheduling, production facility upgrades, buy/make decisions, concept sketches, and so on.

Design

Design has the responsibility of translating the market-driven design specifications derived from product planning into geometrical and topological representations. These representations and their assembly describe the final system and eventually its function. They may use manual drafting, computer-aided design (CAD), 3D CAD, solid modelling tools like rapid prototyping (RP) models, and more.

Much research has been made in this field and numerous tools developed. An example could be the cooperative ARCADE (Automatic Radiosity for Complex and Dynamic Environments) system developed by [IGD, 2004]. It enables several users to model in a shared distributed virtual 3D design space simultaneously. The direct manipulations carried out by the various users are displayed in real-time to the other participants. Dynamic access control ensures that each single object is only manipulated by one user at a time. Once the manipulation is completed, the object is accessible to any participant. Avatars (virtual representations of the remote partners) and an integrated video-conference inform the user where the other conference members are located in the 3D design space and from which perspective they are viewing the model. The ARCADE instances communicate via compact messages over the Internet.

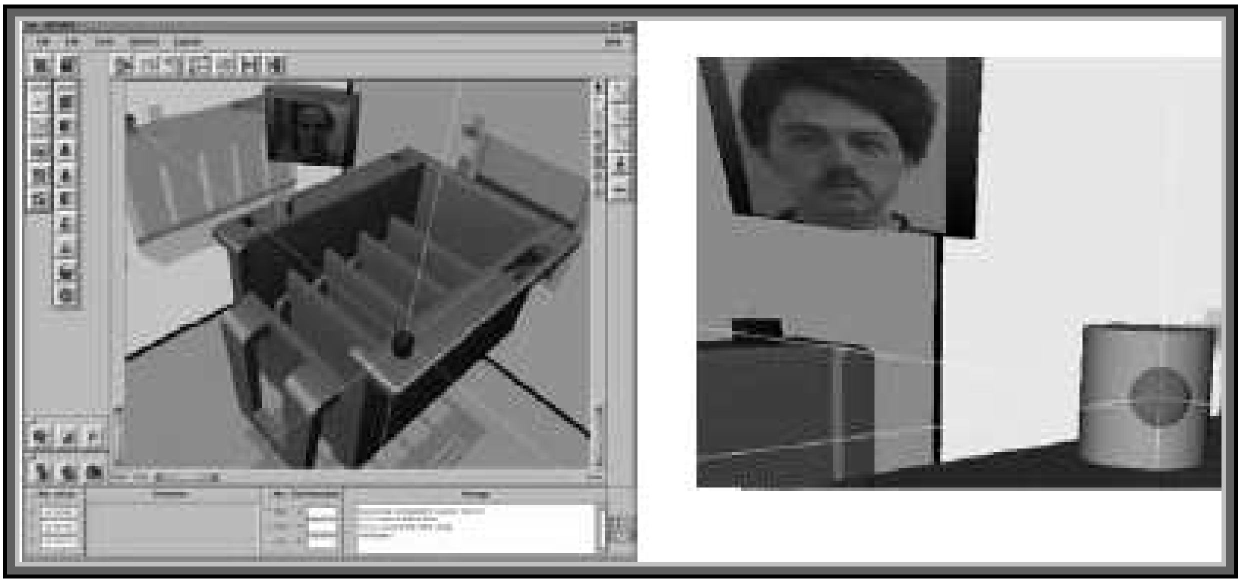


Figure 3-1: Modeling in cooperation, views from the local user and the extern user (cf. [IGD, 2004]).

One of the most interesting examples of distributed collaborative engineering in the field of design is the Interdisciplinary Communication Medium (ICM) developed by the Department of Civil Engineering at Stanford University (cf. [Fruchter et al. 1995]) because it supports the design of complex mechatronic systems, i.e. design in a cross-domain engineering context. In order to facilitate effective communication across disciplines, a framework enables sharing and capturing multi-criteria design proposals, design semantics, critique, explanation and notifications. ICM integrates a shared graphical modelling environment and network-based services, including tools to critique the performance of the designed system. The technical concept is to offer a graphical design environment which can also be used as an interface between designers as well as toolbox and service to support interdisciplinary design. The improvement of ICM is that it creates a space for a multi-criteria evaluation which in turn facilitates a detailed discussion by specific knowledge-based experts for critiquing mechatronic systems.

Research & Development

Research & Development has the responsibility for ensuring the engineering functionality of the products in consideration. They will also most likely become a focal point in the convergence of various elements of the lifecycle data, each of which would be carried out by expert analysts and engineers. Such studies might include structural and thermal analysis, fluid flow analysis, failure mode analyses, manufacturability, recyclability, tests run on prototypes, mechanical tests, diagnosis and voltage tests, and so on. It may include 3D part and assembly CAD data, bill of materials, computational analysis models, fault tree, cause and effects diagrams, tolerance data, surface specification, tests results, and so on.

For example, in 2002 the Lulea University of Technology in Sweden introduced in 2002 a framework for distributed winter testing based on distributed engineering tools combined with telematics (cf. [Jeppson et al., 1999]). The Distributed Interactive Virtual Environment (DIVE) from the Swedish Institute of Computer Science (SICS) supports this framework. This experimental platform for the development of virtual environments, user interfaces and applications based on shared environments is a multi-user application, in which participants interact over a computer network. DIVE enables engineers from different sites with different systems to interact and cooperate.

Forward/Value Engineering

This group is charged with advanced research and development activities in search for material, design practices, and manufacturing processes that will yield a greater value than those currently in use. They are searching for new concepts which can be applied to the future product programs. Toward this end, they work closely with the design and engineering staff. Consequently they also desire to share data with the project-specific design, engineering and manufacturing groups.

Manufacturing

This group develops the manufacturing and assembly process. They will probably have part of the responsibility for the implementation of quality controls, including process monitoring effort. Toward this goal they may use computer-aided manufacturing (CAM) tools, such as numerical control tools (NC-Tool), simulation, robotic simulation, computer-aided process planning (CAPP) software, measuring machines, inspection programs. Due to the variety of data systems, their work can also be supported by a collaboration platform.

Production Planning and Management

This group is in charge of the planning of all resources, for example, facilities, materials, human, transportation, etc. for the actual production. They plan, purchase, and upgrade the resources, work with methods like Just-In-Time and so on. To develop some production schedules, they require a collaborative environment to share data like Bill of Materials (BOM), inventory schedules, facility layout plans, purchase orders, manufacturing resources planning, material requirements planning and so on.

These groups must come together and work collaboratively to create the optimum life-cycle-based and process design possible. Considering these strategic development groups and the quantity of information they share to develop a market-oriented product, it’s evident that a collaborative engineering environment is extremely important. Such an environment could be used to support management tasks, communication, decision-making, group-thinking, testing, validation, planning, certification, and much more.

Before starting the implementation of collaborative engineering tools, it’s necessary to take into account the context of a particular organisation. Every organisation is unique. For this reason the facts and opinions surrounding one organisation’s collaborative engineering initiative will be entirely different from those of others. Careful attention should be given to the chosen solution. Therefore, let us observe some organisational and technical considerations.

However, it is undeniable that organisations won’t be ready for collaborative engineering until they acknowledge the mentality required to encourage shared interaction within teams and discourage isolationism. Collaborative engineering is fundamentally based on the methodology of concurrent engineering. If the organisation has already made an attempt at concurrent engineering and succeeded, then collaborative engineering will probably succeed, too. In the other case it has to plan the initiative much more carefully (cf. [Mills, 1998]). Various influencing factors must be taken into account when planning a collaborative environment. They consist principally of factors of distribution and interdisciplinarity. These factors have already been described in chapter 2. According to [Mills, 1998], the corporate culture is of great importance. The organisation must select values and opinions in order to define what it feels might be the best approach. The organisation must also determine to what degree the people are dispersed by the corporate structure. In fact, the physical or organisational dispersion will influence the appropriate plan for a new collaborative environment. A collaborative engineering initiative implicitly assumes, for the most part, that the organisation is using team-based structures. Another factor is the capacity of the practices and enabling technologies of the organisation to support an integrated enterprise, that is to say any practice or supporting technology that promotes the increased integration of an enterprise may be an important element.

3.2. Classification of Standard IT-Tools to Support Collaboration Processes

Toward this end much previous research can be used, such as the classification schema for groupware, CSCW-applications of [Teufel et al., 1995], or the formalization of CSCW-applications of [Chang et al., 2001]. One of the most famous classification schemas is the time/space matrix of [Johansen, 1988], which classifies groupware in the matrix shown in Figure 3-2. Since we are trying to configure IT-platforms regarding its functionality, we have chosen to classify the standard IT-tools or groupware by the function of each tool like [Ellis et al., 1991] or [Teufel et al., 1995] did, as these seem to be the most appropriate classification for our problem.

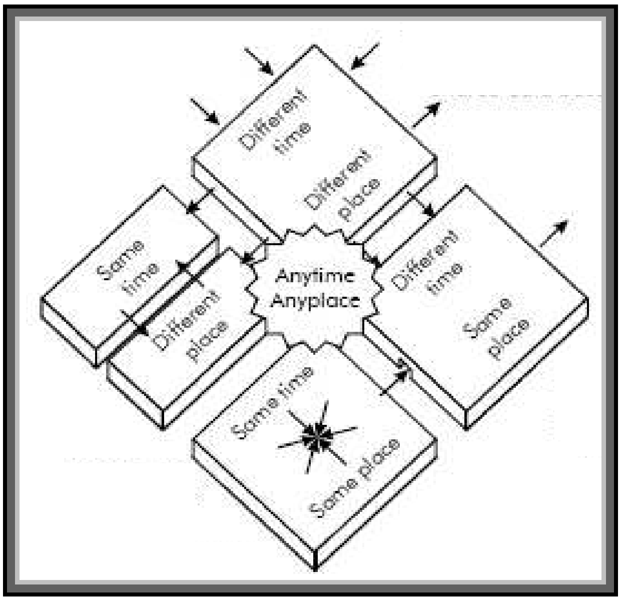


Figure 3-2: Time/space matrix.

[Teufel et al., 1995] classifies groupware in 3 categories, which are principally tasks-oriented: C*ommunication*, C*oordination* and *Cooperation*. They are hierarchically dependent on each other. For example, cooperation processes, like the creation of a business plan, need coordination processes (e.g. planning of review meetings). Coordination again requires adequate communication processes (to send, for example, propositions of business plans to each business unit). We have extended these 3 groups to the notion of collaboration in order to consider in our classification the solutions presented in paragraph 3.1.2.

The hierarchical disposition of these groups is shown in Figure 3-3.

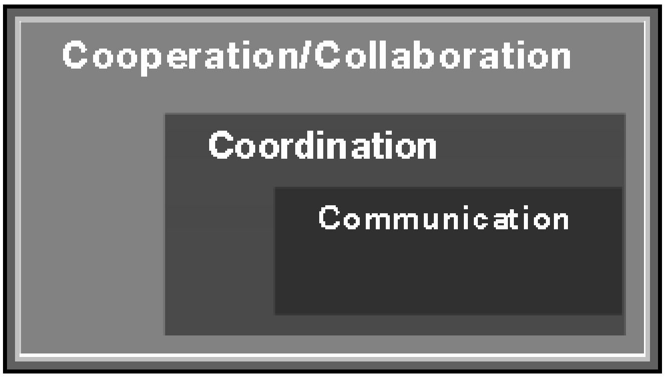


Figure 3-3: Classification groups for IT-tools.

3.2.1. Communication

Communication can be defined as a process by which information is exchanged between individuals through a common system of symbols, signs, or behaviours. According to [Brehmer, 1991], communication is the "life blood" of the organisation, and the greater the need for coordination and cooperation, the greater the necessity for communication.

Typical CSCW-applications for supporting communication processes are, for example, conferencing systems, E-Mails, whiteboards, etc.

Figure 3-4 shows the position of the category C*ommunication* in the classification of standard IT-tools (cf. [Teufel et al., 1995]).

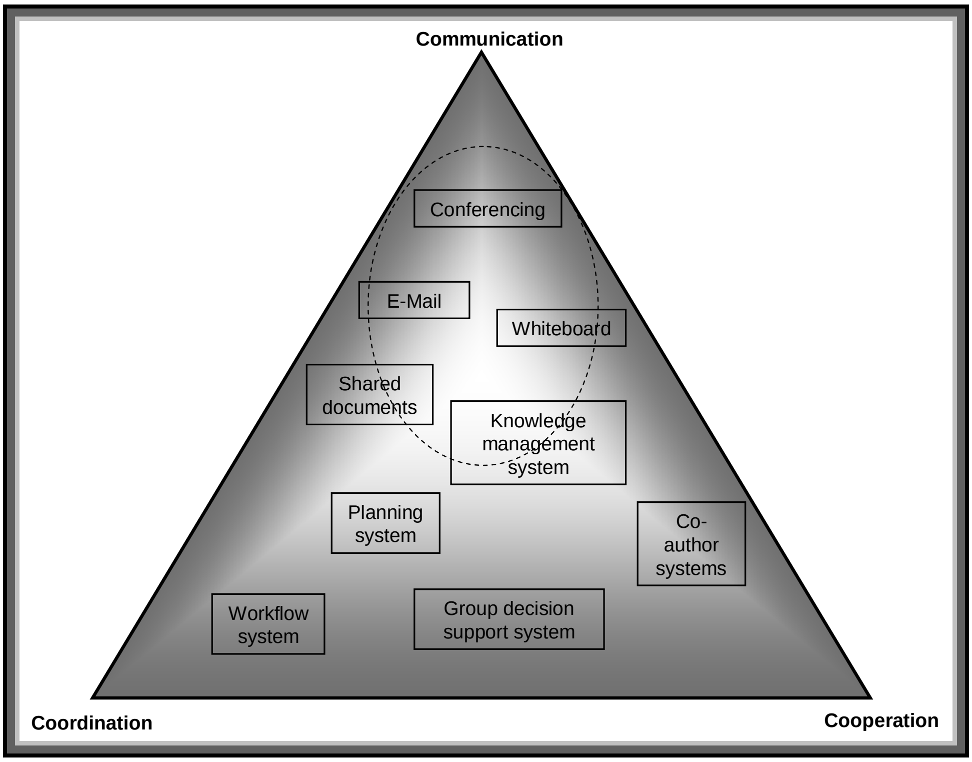


Figure 3-4: Classification of the category Communication (in accordance with [Teufel et al., 1995]).

3.2.2. Coordination

Coordination is the organisation of efforts of different parties to reach a common goal. High-stakes issues are not often involved, and parties need not extend a relationship beyond the accomplishment of the task at hand (cf. [Webster, 2002]). According to [Bordeau et al., 1997] coordination can be defined as the management of dependencies between activities and the support of dependencies among actors. Its purpose is to avoid gaps and overlap in individuals' assigned work. Optimally coordination processes harmonise tasks, roles and schedules in simple environments and systems with the help of project management tools.

Typical CSCW-applications for supporting coordination processes are, for example, workflow systems, planning systems, etc.

Figure 3-5 shows the position of the category C*oordination* in the classification of standard IT-tools (cf. [Teufel et al., 1995]).

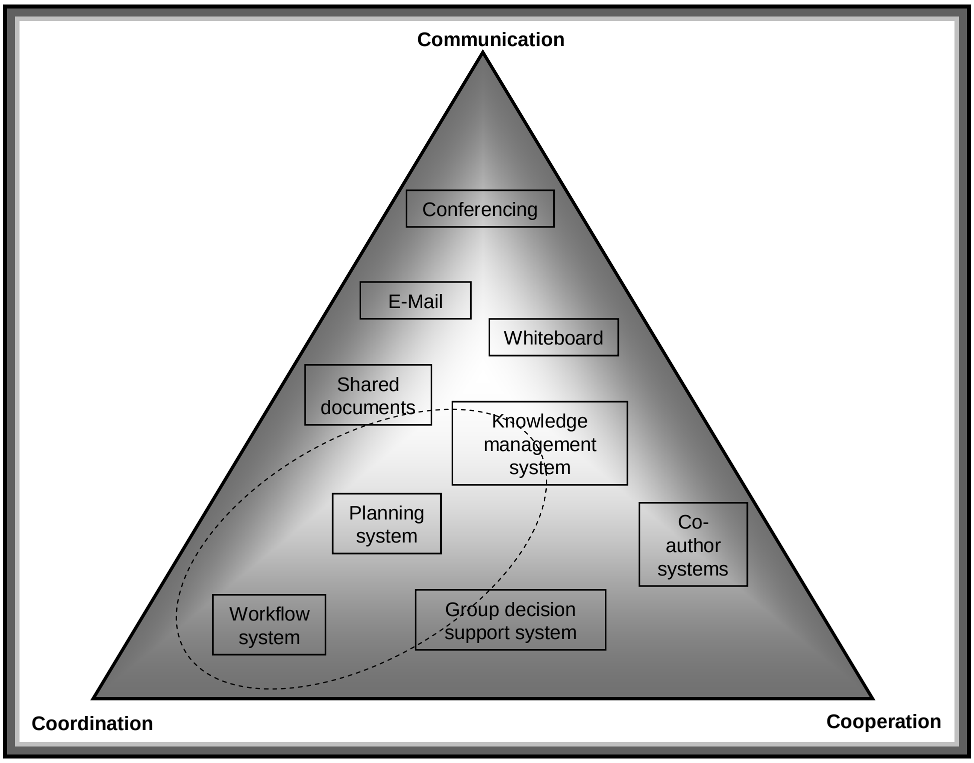


Figure 3-5: Classification of the category Coordination (in accordance with [Teufel et al., 1995]).

3.2.3. Cooperation/Collaboration

Cooperation is a means to an end that involves gains and losses on the part of each participant. This can sometimes foster a competitive environment, and parties need not extend a relationship beyond the accomplishment of the task at hand. In cooperation the participants obtain mutual benefit by sharing or partitioning work. They try to solve problems in complicated environments and systems. The degree of interdependence in designing the effort's work-products is considerable in such projects (cf. [Webster, 2002]).

Typical CSCW-applications for supporting cooperation processes are, for example, co-author system, group decision support systems, etc.
Figure 3-6 shows the position of the category C*ooperation* in the classification of standard IT-tools (cf. [Teufel et al., 1995]).

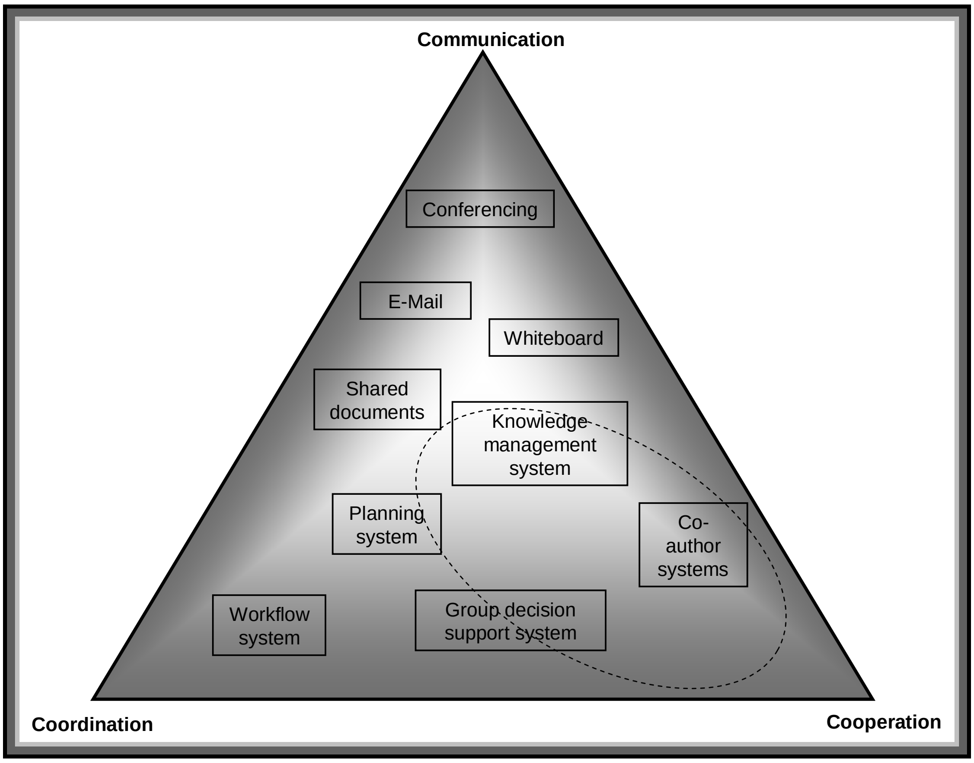


Figure 3-6: Classification of the category Cooperation (in accordance with [Teufel et al., 1995]).

In the category *Cooperation* we have introduced the concept of collaboration. According to [Dillenbourg et al., 1995], cooperation and collaboration do not differ in terms of whether or not the task is distributed, but by virtue of the way in which it is divided; in cooperation the task is split hierarchically into independent subtasks; in collaboration cognitive processes may be heterarchically divided into intertwined layers. In cooperation, coordination is only required when assembling partial results, while collaboration is a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem.

Introducing the category *Collaboration* permits the entry of collaborative engineering tools in our classification (see §3.1.2). We distinguish between discipline specific tools like CAD (Computer Aided Design), CAM (Computer Aided Manufacturing) -systems, BOM (Bills of Material), etc. and cross-discipline tools like requirement and change management systems or an online-translation tool.

3.3. Considered Repertory of IT-Tools

Classifying IT-tools in the above mentioned categories gives us a adequate overview of IT-tools covering the major processes of teamwork. It allows us to create a repertory of IT-tools supporting CEE and CDE contexts. This repertory is presented in Table 3-1 and has been established with the help of the classifications of [Teufel et al., 1995], [Ellis et al., 1991], [Johansen, 1988] and [Wilson, 1991].

E-Mail system	Electronic mail, abbreviated e-mail or email, is a method of composing, sending, storing, and receiving messages over electronic communication systems.
Instant messaging	Instant messaging requires the use of a client program that hooks up an instant messaging service and differs from e-mail in that conversations are then able to happen in real-time. Most services offer a presence information feature, indicating whether people on one's list of contacts are available to chat (e.g. Messenger, ICQ, AOL Buddy-List, etc.).
Online-Conferencing	Online-conferencing is used to hold group meetings or live presentations over the Internet. The most basic feature of a web conference is screen sharing. This is usually accompanied by voice communication, either through a traditional telephone conference or through VoIP, although sometimes text chat is used in place of voice.
Chat room	A chat room is an online site in which people can talk by broadcasting messages to people on the same site in real time. Sometimes these venues are moderated either by limiting who is allowed to speak or by having moderation volunteers patrol the venue watching for disruptive or otherwise undesirable behaviour.
Whiteboard	The term whiteboard is used metaphorically to refer to features of computer software applications that simulate real whiteboards. Virtual whiteboards allow one or more people to write or draw images on a simulated canvas. Each user connects to the whiteboard and can see what other users are writing or drawing in real-time on their computer screen.

Forum	A forum is a platform to facilitate and manage discussions continually among team members. Each thread entails a discussion or conversation in the form of a series of member-written posts. These threads are saved on the forum website for future reading indefinitely (not real time like a chat room) or until deletion by a moderator.
Document management system	A document management system is made up of software designed to manage all types of documents, including scanned, electronic, and paper. All documents are stored in a single repository that facilitates all actions that need to take place from search and retrieval to email and printing.
Knowledge management system	A knowledge management system (KM system) collects, organizes, manages, and shares various forms of information; it is a distributed hypermedia system for managing knowledge in organisations, supporting creation, capture, storage, and dissemination of expertise and knowledge. The idea of a KM system is to enable employees to have access to the organization's knowledge of facts, sources of information, and solutions. A KM system could be document-based, ontology-based or based on AI technologies which use a customized representation scheme to represent the problem domain (cf. [Jennex, 2005].
Electronic calendar	An electronic calendar schedules events and automatically notifies and reminds project members.
Project management system	Project management system is a term covering many types of software, including scheduling, resource allocation, communication and documentation systems, which are used to deal with the complexity of large projects. It is principally used to schedule, track, and chart the steps in a project as being completed.
Workflow system	A workflow system is helping organisations to specify, execute, monitor, and coordinate the flow of work cases within a distributed office environment. It is a collaborative management of tasks and documents within a knowledge-based business process.
Social software system	A social software system organizes the social relations of Group (e.g. Social Network Search Engines, Virtual Presence, Awareness, Mailing Lists, etc.). It reflects by design the traits of social networks and is designed very consciously to let social network analysis work with a very compatible database. All social software systems create links between users, as persistent as the identity those users choose.

Problem solving system	A problem solving system is a software-supporting problem solving processes and creative work (e.g. Brainstorm, Mind Mapper, Thinkature, etc.). It often uses problem-solving techniques like brainstorming, morphological boxes, the method of focal objects, or lateral thinking.
Co-Author system	The terms co-authoring or collaborative writing refer to projects in which written works are created collaboratively by multiple people together (collaboratively) rather than individually (e.g. wikipedia).
Online-Translation	Online-translation is a form of translation wherein a human translator translates texts using computer software designed to support and facilitate the translation process.
Requirement, change management tool	These tools support requirement, change management and configuration management activities (e.g. DOORS, SCM, etc.).
Discipline specific tool	See paragraph 3.1.2. e.g.: CAx-tools, PDM-systems (Product Data Management), Electronic Resources Planning Systems (ERPS), Production Planning Systems (PPS), Simulation (VR, Virtual Reality), Testing Systems, etc.
Electronic Meeting System (EMS), Group Decision Support System (GDSS)	An EMS or GDSS is a suite of highly-configurable collaborative software tools that can be used to create predictable, repeatable patterns of collaboration among people working toward a goal. People can contribute anonymously to most electronic meeting systems tool and so the system provides an equal opportunity for participation. Typical tools of an EMS: Electronic Brainstorming (Problem-Solving), Categorizer, Group Outliner, Rank Order Vote, Alternative Analysis, Topic Commenter.

Table 3-1: Repertory of IT-tools.

4. CONFIGURING IT-PLATFORMS FOR COOPERATION PROJECTS

The goal of this thesis is to develop a method which enables the configuration of IT-platforms for cooperation projects, cross-enterprise or cross-domain ones. The previous chapters have presented, firstly, characteristics and problems of these engineering projects and, secondly standard IT-Tools for supporting such activities. This chapter is the main part of the thesis and describes the method developed.

4.1. The Approach

Principally, the method to be developed aims at optimising decision-making in the IT at the beginning of cooperation projects. For this kind of question, research in social fields such as information management, project management, communication or psychology, is indispensable. In this section we present and evaluate the most common methods to solve this kind of problem: experimentation, modelling, analysis, analogy and reasoning.

1. Experimentation

From a scientific point of view, an experiment is a set of observations performed in the context of solving a particular problem or question to support or falsify a hypothesis or research under well defined conditions. In our context, it means that we have to pilot the IT-tools under consideration in each particular form of cooperation project. As we have characterised a cooperation project by 16 factors having each two values or more, the experimentation may be very time-consuming and expensive. In fact, we would first have to find for each particular project profile a cooperation project in order to pilot the IT-tools in their context, and then observe their work with the IT-tools, which may last some weeks or months. This is a very expensive and laborious process.

Another possibility to use experimentation to solve our problem is to use the sampling method or case studies. Rather than using large samples and following a rigid protocol to examine a limited number of variables, case study methods involve an in-depth, longitudinal examination of a single instance or event: a case. A common understanding about case-study research is that one cannot generalise from a case study. However, through information-oriented sampling of cases one may arrive at case studies that allow generalisation. Nevertheless to solve our problem the number of case studies needed to generalise is certainly greater than ten[2], which is already a great number of projects to carry out a pilot with all IT-tools into consideration. Therefore, case studies are inappropriate to solve our problem.

2. Simulation/Modelling

A simulation is an imitation of some real thing, state of affairs, or process. In scientific research models as representation of the real phenomenon or system are frequently in use. Key issues in simulation include acquisition of valid source information about the object of investigation, a selection of its key characteristics and behaviours, the use of simplifying approximations and assumptions within the simulation, and fidelity and validity of the simulation outcomes. In the case of social and problem-solving research, the most common modelling technique consists in a morphological analysis, a method developed by Fritz Zwicky (cf. [Zwicky, 1969]) for exploring all the possible solutions to a multi-dimensional, non-quantified problem complex.

As a problem-structuring and problem-solving technique, morphological analysis is designed for multi-dimensional, non-quantifiable problems where causal modelling and simulation do not function well or at all. Essentially, morphological analysis is a method for identifying and investigating the total set of possible relationships or configurations contained in a given problem complex in modelling the problem into a morphological box. Then, the morphological box or multidimensional matrix, which contains all of the potential solutions of the given problem, is constructed, and subsequently, the solutions are closely scrutinized and evaluated with respect to the purposes that are to be achieved (cf. [Ritchey, 1998]). This method helps especially in structuring the given problem and its possible configurations, as well as identifying eventual contradictions. However it does not provide any method to find out solutions, thus we could use this problem-solving technique to identify unworkable configurations of cooperation projects, but not to solve our problem.

3. Analytical Method

The analytical method consists of dividing a complex problem into simpler sub-problems in order to find a solution first to the sub-problems, and subsequently to solve the general one. In our case, it is common to divide the problem into the different processes followed during the cooperation project.

In fact, most of the methods providing advices about implementing IT-tools are based on precisely defined and implemented processes and tasks, as mentioned in section 1.4. In the majority of cases an analysis of the existing processes is carried out by an analyst or expert, which then associates IT-tools to the defined processes, as for example in the approaches of [Zizala, 2006] or [Schrötter, 2002]. However, we believe that the selection of collaboration-tools is not really depending on the processes followed in the project as has also been pointed out by a recent study of the Aberdeen Group about Product Lilfecycle Collaboration which states that collaboration isn’t a process in itself (cf. [Brown, 2006]). We discuss this point of view in the following paragraph.

[2] In fact, in [Laurent, 2007a], we already identified at least seven rough forms of cooperation projects.

Probably the most important argument not to base our methodology on processes is that the processes do not really have relevance on using one or another of the IT-tools into consideration, but intervene in a later phase of the project. In fact, with this methodology, we want to give advice in the very-early phase of a project, which IT-tools can be used or not, so that project leaders can, for example, evaluate their IT-costs at the beginning of the project or that the participants don't lose time by having no IT-support at all. The methodology gives advice about the IT-tools that probably will work in the project, regarding the participants, the type of project, the cooperation form, etc. After this, processes come into consideration to define how and when the participants have to use the proposed functionalities. E.g. the method recommends using a workflow-system. After studying the processes in the considered project or defining new processes, the project leader can tell in which cases workflows have to be started, by which participant, with which frequency and which priority. Processes intervene after selecting the IT-tools that actually come into consideration. The methodology we want to develop is concerned with the selection of the IT-tools before the processes ever come into consideration.

Another reason not to base our method on processes is that the development of this method has been motivated through many requests for developing collaboration-platforms within an industrial context, where processes can vary and are often intransparent.

In fact, project leaders from different fields of work and business units following different processes and executing various tasks, requested new IT-support for their cross-domain and/or cross-enterprise projects. As already mentioned in section 1.2, choosing the most appropriate IT-support or configuring an adequate IT-platform is quite difficult and can be a very time-consuming process with costly analyses. Toward this end we decided to automate this decision process by providing a new methodology.

Taking the industrial context into consideration, it does not make sense to develop a methodology dependent on processes or tasks to execute, since we would have to adapt our method to the processes of the considered field of work each time we want to apply it. We need a process-independent method that can be easily applied to each kind of engineering projects: software engineering, mechanical engineering, industrial engineering, systems engineering, civil engineering, and combinations of them.

Moreover, engineering projects are not automatically based on one precisely defined process, but most likely a combination of various processes. In fact, in a complex organisation, projects may follow at the same time various processes, which would require a time-consuming analysis of the project in consideration. We also possibly want to apply our method to projects which do not have any precisely defined and implemented processes yet, as in the case of innovative engineering projects.

These considerations led us to refer to other criteria than the process such as the cooperation form, the global project profile, or the project context. In [VDA, 2001], the method is principally based on the form of cooperation. They classify first the project regarding this criterion, and then use the classification and a process-oriented approach to determine the prospective IT-costs. We consider the cooperation form, as well as the project profile and context, in order to provide a more global and versatile method.

4. Analogy

Analogy consists in the process of solving new problems based on the solutions of similar past problems. For problem-solving applications analogy is also often known as case-based reasoning. Analogy, or case-based reasoning, has been formalized for purposes of computer reasoning as a four-step process:

- Retrieve: Given a target problem, retrieve cases from memory which are relevant to solve it.
- Reuse: Map the solution from the previous cases to the target problem.
- Revise: Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise.
- Retain: After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory.

However, it is quite difficult to find enough cooperation projects in the past that met the same problems and difficulties. In fact, cooperation projects are recent forms of engineering projects that are very complex, meeting each time new difficulties depending on the other cooperation partner, their distance, their form of cooperation, etc. Moreover, cooperation projects in the past didn't have so many or equivalent IT-tools at their disposal, which makes a method based on analogies inappropriate to solve our problem.

5. Reasoning

Reasoning is the mental (cognitive) process of looking for reasons, beliefs, conclusions, actions or feelings (cf. [Kirwin, 1995]). Cognitive science sees reasoning analog to data processing, where relations between observed properties are used in numerous models leading to evident logically correct conclusions in different circumstances. Therefore, it is also an inevitable component of cognitive decision-making, which often uses computational models, such as knowledge-based systems or expert-systems.

The key idea in expert systems technology is that problem solving is accomplished by applying specific knowledge rather than specific technique. It reflects the belief that human experts do not process their knowledge differently from others, but they do possess different knowledge.

In principal, an expert-system is a computer program that contains domain-specific knowledge of one or more human experts. Every expert-system consists of two principal parts: the knowledge base and the reasoning, or inference engine (cf. [Buchanan et al., 1988]).

The knowledge base of an expert-system contains both factual and heuristic knowledge. Factual knowledge is specific knowledge that is widely shared, typically found in books or journals. In contrast, heuristic knowledge is mostly individualistic. It is the knowledge of good practice, good judgement, and plausibility (cf. [Engelmore et al., 1993]).

The problem-solving model organizes the steps taken to solve the problem. One common, but powerful, paradigm involves chaining of IF-THEN rules to form a line of reasoning. Problem-solving methods are built into program modules called inference engines or inference procedures that manipulate and use knowledge in the knowledge base to form a line of reasoning.

Because an expert-system uses uncertain or heuristic knowledge (as we humans do) its credibility is often in question (as is the case with humans). When an answer to a problem is questionable, we tend to want to know the rationale. If the rationale seems plausible, we tend to believe the answer. So it is with expert-systems. The most important ingredient in any expert-system is knowledge. The power of expert-systems resides in the specific, high-quality knowledge they contain about task domains (cf. [Engelmore et al., 1993]).

Since we need a method, which can be used promptly and doesn't require any profound analysis of the project, this method seems to be the most appropriate. In fact, it may take time to set up an expert-system because of the amount of data to collect and the creation of logical rules, but it is generally very simple to apply, since the user only has to answer some questions in form, for example, of a checklist and directly gets a proposition or solution. The advantage of using checklists is that they can be done individually and forces the user also to think about factors that otherwise might have been neglected or forgotten. With an expert system, the expertise gained over many years is made available to less experienced practitioners, which may not have the capacity of analysing the project in detail by themselves.

As mentioned above, an expert-system is based on expert-knowledge, as well as inference rules. The most common way to create inference rules is to use classical predicate logic. The language of classical predicate logic has sufficient expressive power for the formalization of most of mathematics. Classical predicate theory consists of a set of axioms. and the statements deducible from them. Predicate logic is a two-valued logic, where statements need to be either true or false.

Fuzzy Logic is an extension of two-valued logic such that statements need not be true or false, but may have a degree of truth between 0 and 1. Such a system can be extremely useful in designing real-systems, where statements are not completely truth or false. The motivation of using fuzzy logic for our expert-system lies in the human nature of assessing any given situation (here the project profile) in an imprecise manner. Such assessments are often given by vague terms such as “maybe” or “somewhat”, which can be modelled in fuzzy logic. This aspect is detailed in section 4.2.2. The use of fuzzy logic also allows an input based on suggestions instead of exact values, which affirms the idea, that such a system does not need any preliminary analysis of the project.

The expert-system which we developed makes propositions or recommendations to configure an IT-platform regarding a specific configuration. The next figure shows the method in principle. It functions like a funnel. The various IT-tools (marked A to R on Figure 4-1) are potential candidates for the collaboration-platform. Once the profile of the project has been specified, the funnel opens up, and allows some IT-tools become part of the platform.

In practice the project's configuration is defined by the project leader. With the help of a checklist he determines a value for each of the factors mentioned in section 2.4.

The chosen IT-tools are determined by the expert-system which uses fuzzy logic, and is then a so-called fuzzy system (see chapter 4.2). Principally, we use fuzzy logic, since most of the influencing factors used as input of the expert-system are human factors and are not strictly true or false (see also section 4.2.2).

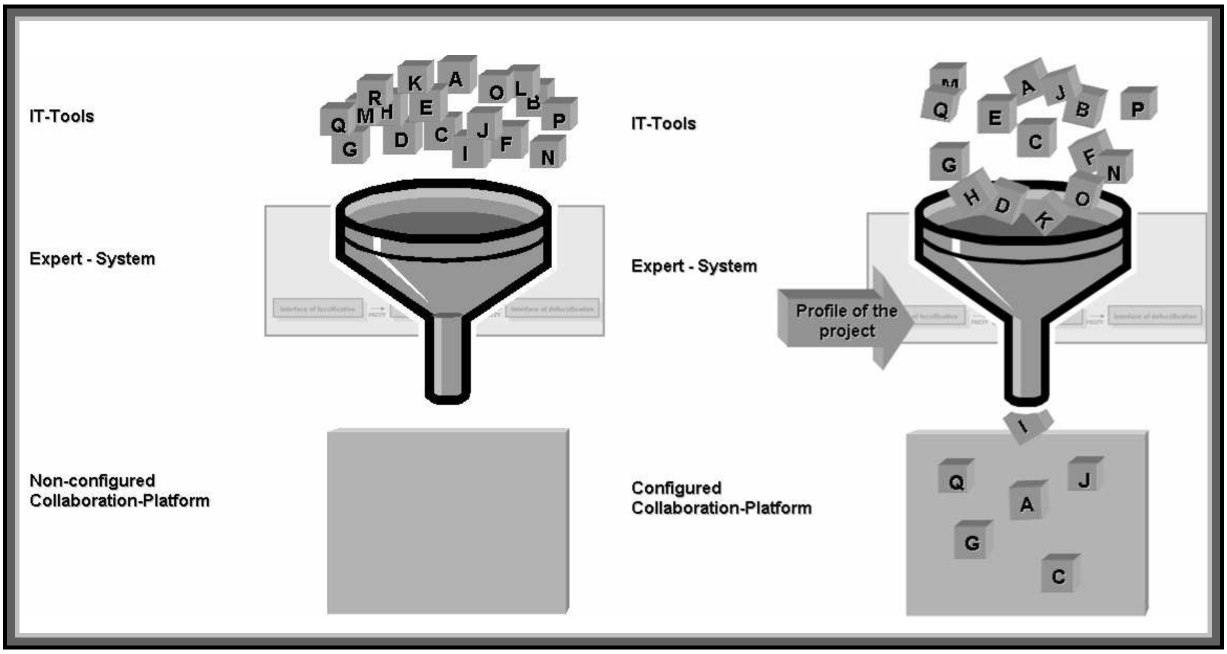

IT-Tools
Expert - System
Non-configured Collaboration-Platform
IT-Tools
Expert - System
Profile of the project
Configured Collaboration-Platform

Figure 4-1: Overview of the expert-system.

The construction of the expert-system, including the followed methods, its inputs, output and implementation are described in detail below. To verify the developed methodology and the associated expert-system we apply it to 3 engineering projects, two product engineering projects and a software engineering project, all presenting aspects of CEE and CDE.

4.2. Fuzzy Logic

Human beings make decisions based on rules, even though we may not be aware of it. For example, if the weather is fine, then we may decide to go out. If the forecast says the weather will be bad today, but fine tomorrow, then we make a decision not to go today and postpone it until tomorrow. Rules associate ideas and relate one event to another. They are also the basis of fuzzy expert-systems. In the following section we introduce first some principles of the fuzzy-set theory and then the creation of rule-bases in fuzzy expert-systems.

4.2.1. Introduction

“So far as the laws of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality.”
Albert Einstein, Geometry and Experience (cf. [Einstein et al., 1921])

Fuzzy logic is derived from fuzzy-set theory dealing with reasoning that is approximate rather than being precisely deduced from classical predicate logic. It can be thought of as the application side of fuzzy-set theory dealing with well thought-out real world expert values for a complex problem (cf. [Klir et al., 1997]). It is important to separate fuzzy-set theory from probability theory, as they are often confused and/or combined: fuzzy logic represents membership in vaguely defined sets, not likelihood of some event or condition (cf. [Mendel, 2001]). The theory of fuzzy sets was introduced in 1965 by Lotfi A. Zadeh in the United States and has been studied and applied by many researchers and practitioners in all parts of the world. More recently, the theory of fuzzy numbers has been introduced by S. Nahmais in the United States and H. Dubois and D. Prade in France. A fuzzy set is a class of objects with grades of membership. Such a set is characterised by a membership (characteristic) function which assigns a grade of membership ranging between zero and one to each object (cf. [Zadeh, 1965]). Fuzzy truth then represents membership in vaguely defined sets, not likelihood of some event or condition and permits the handling of imprecise notions. [Zadeh, 1965] introduced this notion of imprecision in the following way: More often than not, the classes of objects encountered in the real physical world do not have precisely defined criteria of membership. Clearly, the ‘class of real numbers which are much greater than 1,’ or ‘the class of beautiful women,’ or ‘the class of tall men,’ do not constitute classes or sets in the usual mathematical sense. Yet, the fact remains that such imprecisely defined ‘classes’ play an important role in human thinking, particularly in the domain of pattern recognition, information, and abstraction.

Fuzzy logic is used directly in very few applications. Some Fuzzy Logic applications include:

- Control (Robotics, Automation, Tracking, Consumer Electronics)
- Information Systems (Information Retrieval)
- Pattern Recognition (Image Processing, Machine Vision)
- Decision Support (Adaptive HMI, Sensor Fusion)

Most applications of fuzzy logic use it as the underlying logic system for fuzzy expert-systems. An expert-system uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data. The rules in a fuzzy expert-system are usually similar to the following:

If x is low and y is high then z = average,

while x and y are input variables, z is an output variable. Low is a membership function (fuzzy subset) defined on x, high is a membership function defined on y, and average is a membership function defined on z.

The antecedent (the rule's premise) describes to what degree the rule applies, while the conclusion (the rule's consequent) assigns a membership function to each of one or more output variables. Most tools for working with fuzzy expert-systems allow more than one conclusion per rule. The set of rules in a fuzzy expert-system is known as the rule base or knowledge base. This process, also called general inference process, will be explained in more detail in section 4.2.4.

4.2.2. Motives of Using Fuzzy Logic

What distinguishes fuzzy-set theory from classical set theory? According to classical set theory, an element either belongs to one set or not. In fuzzy-set theory an element may also belong partially to a set. Fuzzy sets have gradations of set membership and blurred boundaries. Classical theory has well-defined set boundaries and membership is as clear as black and white. At issue is clarifying to which set gray belongs. Although classical theorists may attempt to create a new set called gray, the problem still persists. When does dark gray become black or conversely when does light gray become white? The fuzzy-theory approach neatly handles the assignment of gray as a partial member of both the white and black sets. (cf. [Treadwell, 1995])

One reason for choosing the fuzzy-set theory is that most of the influencing factors used as input of the expert-system are human factors and are not strictly true or false. Subsequently, they raise the same problems mentioned above.

Take e.g. the factor ‘Skill level in the agreed language’ (factor 3) (see Table 2-3, on page 21): in most cases a team is neither excellent, nor totally bad in the project language. For this reason we need intermediate qualifiers to determine the team’s level with respect to a common project language.

Another reason to use the fuzzy-set theory is that our goal is to configure a collaboration-platform adapted to engineering projects. An important point is that even if the collaboration platform is very efficient and corresponds to the project’s needs, its success depends on the users, their acceptance, the way the new tool is introduced and, most importantly, the new business and engineering processes to be followed. For this reason using soft computing in general and fuzzy logic in particular is an excellent choice. In fact factors like acceptance, usability, and, generally speaking every factor depending on human beings, their culture or discipline, are difficult to examine with first-order logic because of their subjective and qualitative character, and hence dedicated to soft computing techniques which resemble human reasoning more closely than traditional techniques based on conventional logical systems, such as sentential logic and first-order logic, or rely heavily on the mathematical capabilities of a computer. [Dubois et al., 1996] also defend this opinion. They wrote that the fuzzy set membership functions are convenient tools for modelling user’s preference profiles and the large panoply of fuzzy set connectives are capable of capturing the various user attitudes concerning the way the different criteria present in his requirements compensate or not.

4.2.3. Foundation

In this section, we will define the following important notions from fuzzy-set theory which are necessary to construct our fuzzy expert system: fuzzy set, fuzzy number, fuzzy interval, LR-Number, and LR-Interval. The following definitions are based on [Zadeh, 1965], [Kaufmann et al., 1991], and [Zimmermann, 1991].

Let us start with the notion of a fuzzy set. In classical set theory a subset A of any referential set E, $A \subset E$ is defined by expressing the fact that elements of E can be characterised in a specific way, for example, all even numbers are a subset of Z.

In a more abstract way we can also define A by applying a so called characteristic function to all elements in E i.e.:

$$A = \{x \in E | \mu A(x) = 1\}, where\, \mu A(x) \in \{0,1\}$$

So for our example before we would have:

$$A = \text{set of even numbers} = \{x \in \mathrm{Z} | \mu A(x) = 1\}, where\; \mu A(x) = \begin{cases} 1, if\;\, x \bmod 2 = 0 \\ 0, otherwise \end{cases},\; \forall x \in \mathrm{Z}$$

In other words, each element $x \in E$ is either in A or not which complies with classical predicate logic. Now with fuzzy logic a predicate does not necessarily have either the truth value 0 or 1. Instead we can ascribe a truth value between 0 and 1 to a certain statement such as “it may rain tomorrow” which is neither completely true nor false. Correspondingly we can now give the following definition of a fuzzy set.

Definition 1. Fuzzy set

Let E be a referential set and let A be a subset of E. A is called a fuzzy set of E if it is defined by a characteristic function $\mu A(x) : E \rightarrow [0,1]$, called its membership function. That is:

$$A = \{x \in E | \mu A(x) \in \;]0,1]\}$$

Throughout the thesis we work exclusively with special forms of fuzzy sets called LR-Numbers (Left-Right-Numbers) and LR-Intervals, which take as their referential set the set R of real numbers, for the following reasons:

- All factors being investigated in our empirical study can be expressed as real numbers. Therefore we can describe the linguistic variable (LV) for each factor used in our fuzzy-expert-system with fuzzy sets based on the referential set R.

- LR-Numbers and LR-Intervals are easy to represent and thus to manipulate. In fact, according to [Zimmermann, 1991], operations with fuzzy sets in general involve rather extensive computations as long as no restrictions are placed on the type of membership functions allowed.[3]

In order to define LR-Numbers, we first introduce the concept of a fuzzy number and of a fuzzy interval. According to [Richei, 1998], we give the following definitions:

Definition 2. Fuzzy number

A fuzzy number is a fuzzy set A based on the referential set R whose membership function $\mu A(x)$ has the following properties:

- $\mu A(x)$ is continuous on R
- $\exists! x \in R$ *s.th.* $\mu A(x) = 1$

Definition 3. Fuzzy interval

A fuzzy interval is a fuzzy set A based on the referential set R whose membership function $\mu A(x)$ has the following properties:

- $\mu A(x)$ is continuous on R
- $\exists!$ a not-empty closed interval $I \subset R$ s.th. $\mu A(x) = 1 \forall x \in I$

Now that we have defined fuzzy-numbers and fuzzy-intervals we are able to give the definition of LR-Numbers and respectively –Intervals. These are special forms of fuzzy numbers and intervals and, as already mentioned above, simplify clearly the work with fuzzy sets. We work in the thesis exclusively with LR-Numbers and -Intervals. These special forms of fuzzy sets are described with the help of specific functions called shape functions. In the case of LR-Numbers and –Intervals, shape functions are used to describe the left (L) and respectively the right (R) part of the sets' membership functions. We give the following definition of a shape function:

Definition 4. Shape function

In the context of fuzzy sets, the function $S : [0, \infty) \longrightarrow [0,1]$ is called a shape function if and only if:

[3] For this reason, Dubois and Prade in [Dubois et al., 1978] and [Dubois et al., 1979] proposed a general algorithm for performing operations on a special type of fuzzy sets called LR-Numbers and respectively LR-Intervals.

$$S:[0,\infty) \longrightarrow [0,1], \text{ with } \begin{matrix} 1.\ S(0)=1, \\ 2.\ S \textit{ is strictly monotone decreasing in } [0,\infty) \end{matrix}$$

LR-Numbers and –Intervals are defined with the help of two shape functions, one describing the left part of the fuzzy set called L and the other one describing the right part of the fuzzy set called R. With the help of these two shape functions, we are now able to give for LR-Numbers and –Intervals the following definitions:

Definition 5. LR-Number

A fuzzy number A is an LR-Number (and respectively a fuzzy number of type LR), if and only if:

$$\mu A(x)=\begin{cases} L\left(\dfrac{a-x}{\alpha}\right) \textit{if } x \le a; \alpha > 0, \\ R\left(\dfrac{x-a}{\beta}\right) \textit{if } x \ge a; \beta > 0. \end{cases}$$

where:

- a is called culminant point of A with $\mu A(a)=L(0)=R(0)=1$,
- α and β are the right and left limits of A,
- $L(u)$ and $R(u)$ are adapted shape functions following Definition 4.
- Denoted by $A=\left\langle a,\alpha,\beta\right\rangle_{LR}$

An LR-Interval is a fuzzy interval of type LR i.e. a fuzzy interval described with the help of shape functions following Definition 4. LR-Intervals are an extension of the LR-Numbers.

Definition 6. LR-Interval

According to [Richei, 1998], the representation of the membership function of an LR-Interval is defined as:

$$\mu[m1,m2](x)=\begin{cases} L\left(\dfrac{m_1 - x}{\alpha}\right) \mid x \le m_1; \alpha > 0, \\ 1 \qquad \mid m_1 \le x \le m_2, \\ R\left(\dfrac{x - m_2}{\beta}\right) \mid m_2 \le \text{x}; \beta > 0. \end{cases}$$

where:

- $L(u)$ and $R(u)$ are adapted shape functions following Definition 4.
- The amplitude $|m_1 - m_2|$ is called the tolerance region of the LR-Interval
- Denoted by: $A = \langle m_1, m_2, \alpha, \beta \rangle_{LR}$

Figure 4-2 gives an example of an LR-Number and an LR-Interval respectively. To sum up, for an LR-Number with a membership function $\mu(x)$, a is the culminant point of the fuzzy set and $\mu(a) = 1.\alpha$ and β are the right and left limits of the fuzzy set.

For an LR-Interval with a membership function $\mu(x)$, for each $x \in [m_1; m_2]$, we have $\mu(x) = 1$. α and β are also the right and left limits of the fuzzy set.

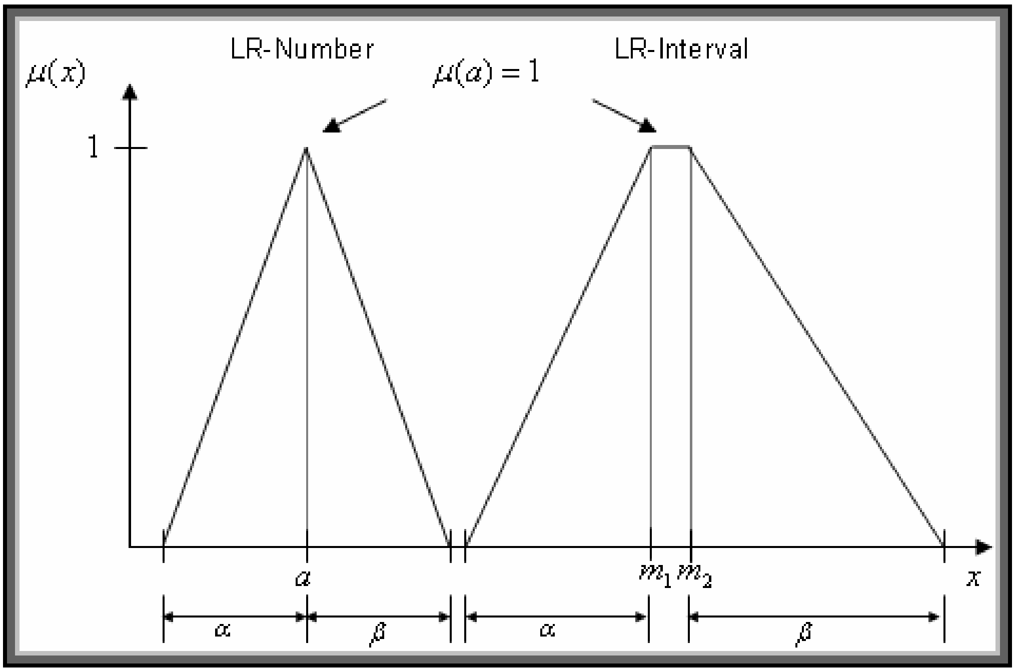


Figure 4-2: Example for LR-Numbers and LR-Intervals (cf. [Richei, 1998]).

Note here that $\alpha \neq \beta$ is possible. If α =0 and/or β =0, a special case of LR-Intervals exists. If α or β is equal to 0, then one of the limits of the fuzzy number or interval is an exact limit. If both are equal to 0, then the notation represents an exact number or interval, or singleton in fuzzy logic (cf. [Richei, 1998]).

4.2.3.1. Linguistic Variables (LV)

The thesis often uses linguistic variables (LV) to facilitate the comprehension of the reader. Just like an algebraic variable takes numbers as values, an LV takes words or sentences as values (cf. [Zimmermann, 1991]). LV are often required, when words as, for example, 'very', 'quite', 'moderately', 'more or less', 'somewhat', 'rather,' or 'sort of', are commonly used to describe the variable. [Zadeh, 1965] also identified this set of natural-language words that he referred to as 'linguistic hedges'.

The set of values that it can take as an input is called its 'term set'. Each value in the term set is a fuzzy set defined by a base (base=referential set). The base of the LV defines the universe of discourse for all the fuzzy sets in the term set.

Definition 7. Linguistic variable (LV)

Let E be a given referential called the base of the LV. Then an LV is defined by a term set T s.th. each element $t \in T$ is described by a fuzzy set F_t over the base E (cf. [Jantzen, 1998]).

Let us consider an example: Let x be the LV 'Human height' with the given term set $T = \{VerySmall, Small, MoreOrLessTall, Tall, VeryTall\}$. Each term of this term set is described by a fuzzy set w.r.t. the base [0,250] cm. For instance the terms 'small', 'tall', and 'very tall' could be given by the following fuzzy sets (see Figure 4-3).

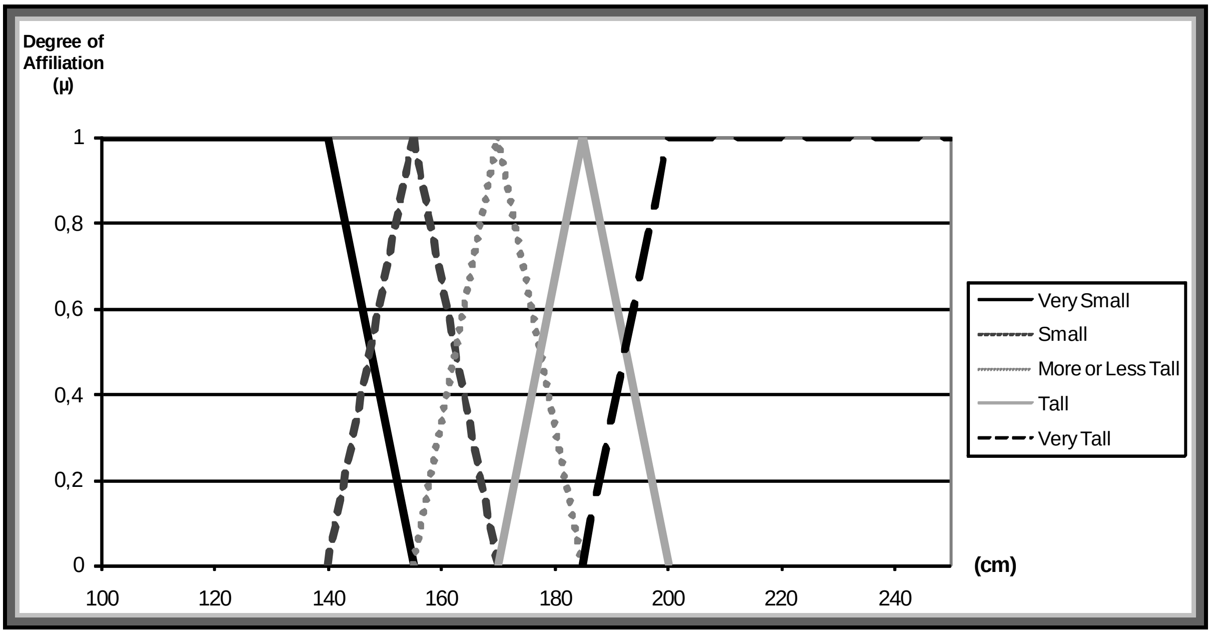


Figure 4-3: Example for the linguistic variable 'Human height'.

4.2.3.2. Consistency of a Linguistic Variable

Linguistic variables, in short LVs, are used in this thesis to define the influencing factors describing a possible project configuration (see Table 2-3).

In principle fuzzy-set theory allows a great many of possibilities to define LVs. However, it also has established axioms to facilitate operations on fuzzy sets as well as their comprehensibility. In the context of this thesis we consider the notion of consistency as an axiom and require our LVs to fulfil the condition of this axiom, since it provides the following advantages.

Consistency guarantees that each possible input for a factor is associated with at least one term of the LV and that each particular fixed input is affiliated with all terms such that the sum of all memberships equals 1. Since the sum of all memberships equals 1, the membership values could be associated with probability values and thus make each particular input more concrete and comprehensible.

Moreover fuzzy-set theory is an extension of classical set theory (cf. [Zadeh, 1965]) which always gives either 0 or 1 as a result of evaluating logical expressions. If we want to mirror the classical set theory with fuzzy-set theory, evaluating logical expressions with the help of fuzzy sets should give results between 0 and 1 which is only possible if the sum of the membership values for an input is not greater than 1.

In the following we describe the conditions an LV has to fulfil to be consistent.

Completeness

Let an LV be given by a term set *T* over a referential set *E* (base) (see Definition 7). Given any value in the base we require that the value is assigned to at least one term $t \in T$ with a certain probability greater than 0. This can be formally written as:

$$\forall x \in E : \exists t \in T \text{ s.th. } \mu_t(x) > 0$$

This corresponds to classical set theory in the following way:

In classical set theory, a given element x of a referential set *E* can either possess a property A or not, which corresponds to the classical two-valued logic. This can be expressed in the following way:

Let a referential set *E* and a term set $T := \{(has\ prop\ A), (\neg\, has\ prop\ A)\}$ be given where $\forall x \in E$, $\mu_t(x)$ is defined as follows:

$$\mu_{has\ prop\ A}(x) = \begin{cases} 0, x \neg\, has\ prop\ A \\ 1, x \quad has\ prop\ A \end{cases}$$

$$\mu_{\neg has\ prop\ A}(x) = \begin{cases} 1, x \neg\, has\ prop\ A \\ 0, x \quad has\ prop\ A \end{cases}$$

Then we have:

$$\forall x \in E:\ \exists!\ t \in T \text{ s.th. } \mu_t(x) = 1$$

To sum up, the notion of completeness can be already applied to classical set theory and extended to introduce the notion of fuzzy-completeness (cf. [Hohl, 2005]).

Partition of unity

In fuzzy set theory we assign membership values to describe whether any given value $x \in E$ is expressed by a term $t \in T$. In the context of probability theory this can be interpreted as assigning a probability value to a certain event. Now we know from probability theory that the sum of probabilities of all possible events must be equal to 1. With an LV this means that:

$\sum_{t \in T} \mu_t(x) = 1$, for any given $x \in E$

In classical set theory this property follows immediately from completeness because $\mu_t(x) \in \{0,1\}$ and $\exists!\, t \in T\ s.th.\, \mu_t(x) = 1, \forall x \in E$.

We now give:

Definition 8. Axiom of Consistency

Let an LV be given by a term set *T* over a referential set *E* (base) (see Definition 7). The LV is called consistent iff:

- $\exists t \in T\ s.th.\, \mu_t(x) > 0, \forall x \in E$
- $\sum_{t \in T} \mu_t(x) = 1,$ for any given $x \in E$

As a result of the advantages mentioned before, we require our LVs to fulfil the conditions of this axiom. This means that each particular fixed input is affiliated with at least one term such that the sum of all memberships equals 1 and must be taken into account by defining our LVs in section 4.3.

4.2.3.3. Operations on Fuzzy Sets

The notions of inclusion, union, intersection, complement, relation, etc., from classical set theory can be extended to fuzzy sets. According to [Zadeh, 1965]), we introduce the following definitions:

Definition 9. Fuzzy-emptiness

A fuzzy set A based on a referential set *E* is called empty if and only if its membership function is identical to zero on *E* i.e. $\mu A(x) = 0, \forall x \in E$.

Definition 10. Fuzzy-equality

Two fuzzy sets A and B based on a referential set *E* are equal, written $A = B$, if and only if $\mu A(x) = \mu B(x)$, $\forall x \in E$.

Definition 11. Fuzzy-complement

The complement of a fuzzy set A is denoted by A' and is defined by $\mu A'(x) = 1 - \mu A(x)$, $\forall x \in E$.

As in the case of ordinary sets, the notion of containment plays a central role in the case of fuzzy sets. This notion and the related notions of union and intersection are defined as follows:

Definition 12. Fuzzy-subset

Given two fuzzy sets A and *B* with the same referential *E*, A is a subset of B if and only if $\mu A(x) \le \mu B(x)$, $\forall x \in E$.

Definition 13. Fuzzy-union

Given two fuzzy sets A and *B* with the same referential *E*, the union of two fuzzy sets A and B with respective membership functions $\mu A(x)$ and $\mu B(x)$ is a fuzzy set C, written as $C = A \bigcup B$, whose membership function is $\mu C(x) := Max(\mu A(x), \mu B(x))$, $\forall x \in E$ point-wise (see Figure 4-4).

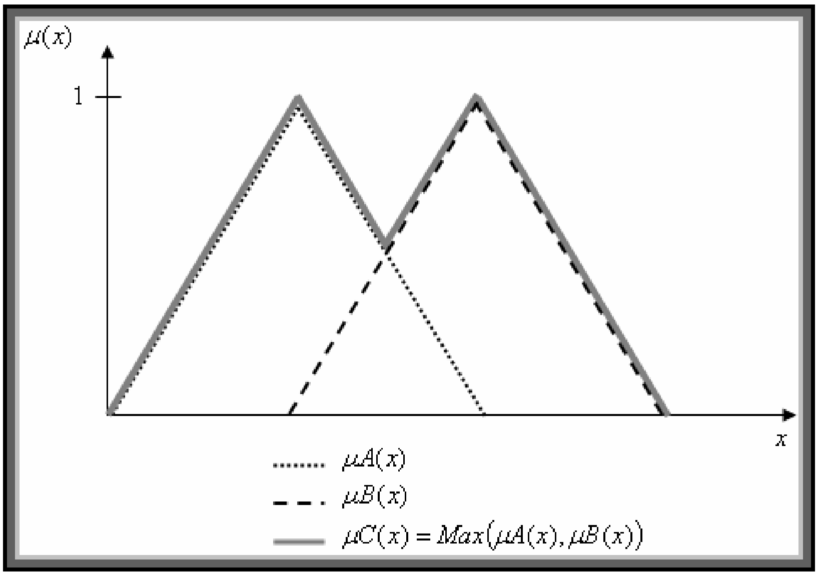


Figure 4-4: Fuzzy-union.

Definition 14. Fuzzy-intersection

Given two fuzzy sets A and *B* with the same referential *E*, the intersection of two fuzzy sets A and B *with respective membership functions* $\mu A(x)$ and $\mu B(x)$ is a fuzzy set C, written as $C = A \bigcap B$, whose membership function is $\mu C(x) := Min(\mu A(x), \mu B(x)), \quad \forall x \in E$ point-wise.

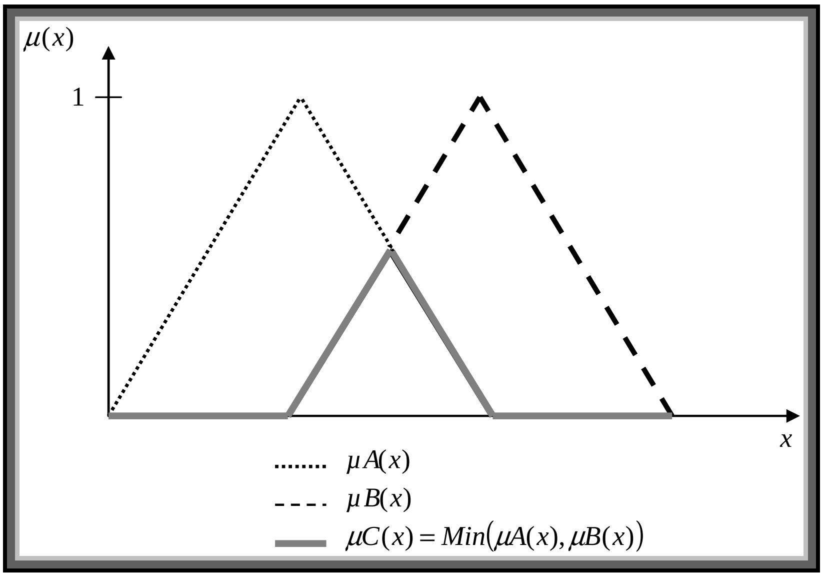


Figure 4-5: Fuzzy-intersection.

Fuzzy-union and intersection have the associative property (cf. [Zadeh, 1965]).

Finally, we introduce the concept of fuzzy relations. Fuzzy relations are fuzzy sets of $X \times Y$, that is, mappings from $X \to Y$. They have been studied by a number of authors, in particular by [Zadeh, 1965], [Kaufmann, 1975], and [Zimmermann, 1991]. A fuzzy relation is obviously a fuzzy subset of the Cartesian product $X \times Y$, denoted as a relation from a set X to a set Y. Their application are widespread and important, therefore we give the following definition:

Definition 15. Fuzzy relation

Let $X, Y \subseteq R$ be universal sets, then $\tilde{R} = \{(x,y) \in X \times Y \mid \mu_{\tilde{R}}((x,y)) \in [0,1]\}$ is called a fuzzy relation on $X \times Y$ (cf. [Zimmermann, 1991]).

4.2.3.4. Rule Base in a Fuzzy-Expert-System

The general idea of a fuzzy-expert-system compared to a classical expert-system is to represent the rule base, already mentioned in section 4.2.1, with the help of LVs. A rule base consists of rules and each rule, in turn, is obtained from properties expressed by linguistic variables and terms and using logical operators (cf. [Schmid, 2005]).

Remember our example in section 4.2.1. The rules in a fuzzy expert-system are usually similar to the following:

“If x is low and y is high then z is average”.

While x and y are linguistic input variables, r is a linguistic output variable. ‘Low’ is a term described by a membership function $\mu_{Low}(x)$ defined on a referential set *X*, ‘High’ described by $\mu_{High}(y)$ defined on a referential set *Y*, and ‘average’ described by $\mu R(z)$ defined on a referential set Z.

The relation between the linguistic terms ‘Low’ and ‘High’, and its representation by the membership function, $\mu R(z)$, is characterised by the property ‘Low’ AND ‘High’. This expression can be written in mathematical form using the elementary connective min operator or fuzzy intersection in two dimensions (cf. [Schmid, 2005]) and illustrated in Figure 4-6:

$$\mu R : X \times Y \to [0,1] \quad s.th. \quad \mu R\big((x,y)\big) = \min_{(x,y)}\{\mu_{\tilde{l}\,ow}(x,y), \mu_{\tilde{H}igh}(x,y)\}$$

With: $\mu_{\tilde{L}ow}(x,y) := \mu_{Low}(x) \otimes y = const$ and $\mu_{\tilde{H}igh}(x,y) := \mu_{High}(y) \otimes x = const$

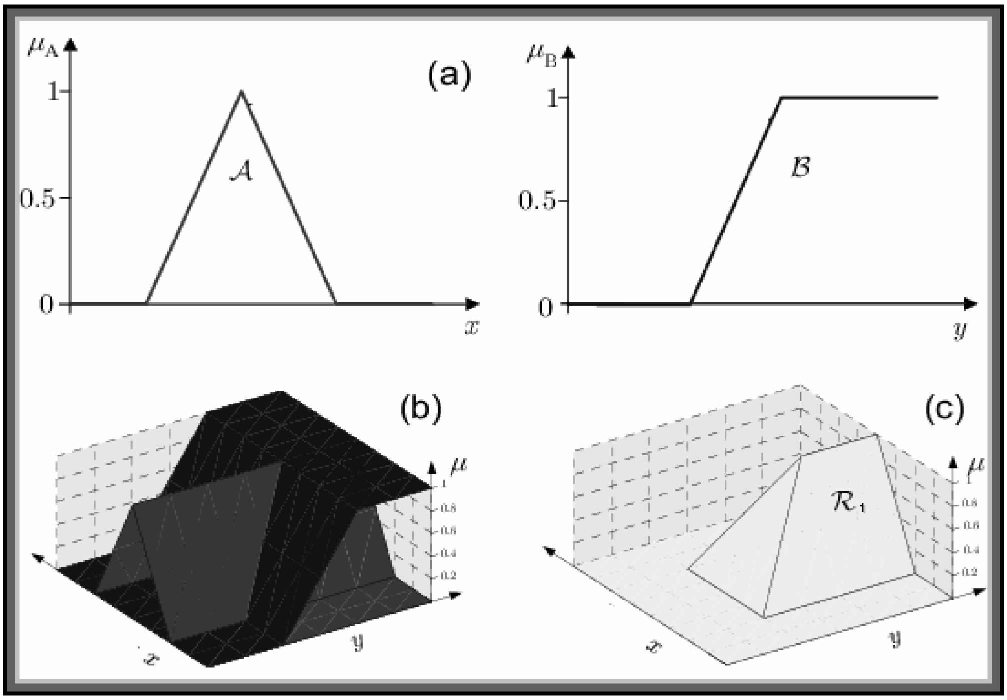


Figure 4-6: Relation between two fuzzy sets: (a) membership functions, (b) 3-D view of the membership functions, (c) membership function of the relation after applying the MIN operation to (b) (cf. [Schmid, 2005]).

Combining rules into a rule base for the same output variable *r*, the example from above could be extended as follows:

(1) If x is ‘low’ and y is ‘high’ then z is ‘average’

OR

(2) If x is ‘high’ and y is ‘high’ then z is ‘high’

OR

(3) If x is ‘low’ and y is ‘low’ then z is ‘low’

Thus, for the same output variable *z* we have 3 rules (in general n rules). The complete rule base represents a union of the existing rules in a linguistic way. The complete rule base is therefore given by combining the fuzzy relations formed for each individual rule with a fuzzy union or max operator:

$$\mu R\ (x_1, y_1, x_2, y_2, x_3, y_3) = \max\{\mu\tilde{R}_1(x_1, y_1, x_2, y_2, x_3, y_3), \mu\tilde{R}_2(x_1, y_1, x_2, y_2, x_3, y_3), \mu\tilde{R}_3(x_1, y_1, x_2, y_2, x_3, y_3)\}$$

With:

- $$\mu\tilde{R}_1(x_1, y_1, x_2, y_2, x_3, y_3) := \mu R_1(x_1, y_1) \otimes (x_2 = const, y_2 = const) \otimes (x_3 = const, y_3 = const)$$
- $$\mu\tilde{R}_2(x_1, y_1, x_2, y_2, x_3, y_3) := \mu R_2(x_2, y_2) \otimes (x_1 = const, y_1 = const) \otimes (x_3 = const, y_3 = const)$$
- $$\mu\tilde{R}_3(x_1, y_1, x_2, y_2, x_3, y_3) := \mu R_3(x_3, y_3) \otimes (x_1 = const, y_1 = const) \otimes (x_2 = const, y_2 = const)$$

Note: R_i represents the fuzzy relations formed for each individual rule *i* where *i=1…n*$_z$ (n_Z the number of rules for the output variable z).

To sum-up, we use the following steps of our methodology to express the rule base (cf. [Schmid, 2005]):

- Representation of all needed membership functions to express the rules: this means that we first have to be acquainted with the description of all linguistic variables necessary for the input and output of our fuzzy-expert-system. In our example this corresponds to the membership functions $\mu_{Low}(x)$, $\mu_{High}(y)$ and $\mu R(z)$.
- Representation of the logical operator: this step is concerned about determining the appropriate logical operator to express a single rule. In our example, it corresponds to the use of the min operator or fuzzy intersection.
- In this step, the representation of the membership functions and of the logical operator is used to get the representation of all the rules in the rule base. This means that all single rules are now defined with the help of membership functions and the logical operators defined in section 4.2.3.3.
- In the last step, we combine the representations of the different rules into a rule base. This combination is also carried out by a logical operator. Generally, and in our example, it corresponds to the use of the max operator or fuzzy union.

The representation of our rule base is described in more detail in section 4.4.3. This representation of a rule base is the standard max-min representation or max-min composition, as defined in [Zimmermann, 1991]:

Definition 16. Max-min composition

Let $\tilde{R}_1(x, y), (x, y) \in X \times Y$ and $\tilde{R}_2(y, z), (y, z) \in Y \times Z$ be two fuzzy relations. The max-min composition $\tilde{R}_1$ max-min $\tilde{R}_2$ is then fuzzy set:

$$\tilde{R}_1 \circ \tilde{R}_2 = \{[(x, z), \max_y\{\min\{\mu_{\tilde{R}_1}(x, y), \mu_{\tilde{R}_2}(y, z)\}\}] | x \in X, y \in Y, z \in Z\}$$

4.2.4. Use of Fuzzy Logic

The fuzzy-expert-system follows a general inference process which proceeds in three (or four) steps as described in Figure 4-7.

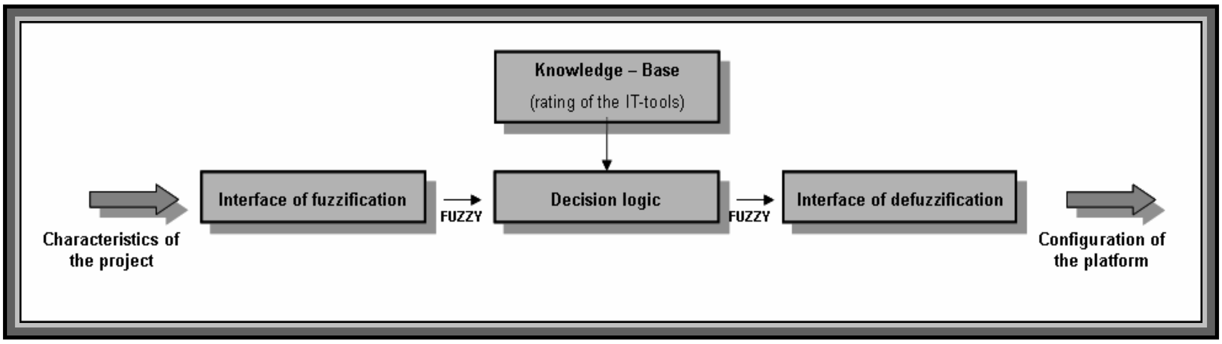

Knowledge – Base
(rating of the IT-tools)
Interface of fuzzification
FUZZY
Decision logic
FUZZY
Interface of defuzzification
Characteristics of the project
Configuration of the platform

Figure 4-7: Fuzzy-System to support decision (according to [Kruse et al., 1995]).

Let us explain the inference process with the help of an example. We consider another time the linguistic variable ‘Human height’ with the term set $T = \{VerySmall, Small, MoreOrLessTall, QuiteTall, VeryTall\}$ defined on a base X with a unit in cm.

4.2.4.1. Fuzzification

During the fuzzification the membership functions defined on the input variables are applied to their actual values to determine the degree of truth for each rule premise. Toward this end the influencing factors have been defined as LR-Intervals or LR-Numbers (see chapter 4.2.3.1). In our case, the interface of fuzzification transforms the characteristics of the project under consideration into fuzzy inputs.

Returning to our previous example: a man does not know if he belongs to the class of ‘more or less tall’ or ‘quite tall’ men. He enters his size and the interface of fuzzification classifies him for example to 20% in the class ‘More or less tall’ and to 80% in the class ‘quite tall’.

4.2.4.2. Decision Logic - Inference

During the inference, the truth value for the premise of each rule is computed, and applied to the conclusion part of each rule. This result is one fuzzy subset that can be assigned to each output variable for each rule. Usually only the operator MIN is used as inference rule. [Mizumoto, 1989] explains that in MIN inference, the output membership function is clipped off at a height corresponding to the rule premise's computed degree of truth. In our system the decision logic (see Figure 4-7) uses the results of an empirical study as knowledge base to determine a fuzzy value corresponding to each IT-tool, which is then compared to the user's inputs.

During the composition, all of the fuzzy subsets assigned to each output variable are combined together to form a single fuzzy subset for each output variable. Again, usually the operator MAX is used. In MAX composition, the combined output fuzzy subset is constructed by taking the point-wise maximum over all of the fuzzy subsets assigned by the inference (cf. [Mizumoto, 1989]).

In our example, the fuzzy-expert-system decides if this man is appropriate, for example, to an open job position. The expert-data states that: if the man is quite tall and his English is very well, then he is appropriate for this job. The fuzzified user input is matched with the rule base in the expert-data. This is carried out by the MIN operator.

Thereafter, the fuzzy-expert-system must consider the entire set of rules corresponding to this open job position and the different factors in order to take its decision. Another rule could be, for example: if the man is very tall and his English is excellent, then he is excellent for this job. This step is carried out by the MAX operator and provides the results in form of an output set representing if the man is appropriate, or not, for the job.

Note that in the thesis the inference is carried out separately for each tool, i.e. we assume that the IT-tools do not interfere with each other. We make this assumption to avoid a too high level of complexity. This means that we repeat the inference process for each considered IT-tool separately.

4.2.4.3. Defuzzification

Finally, the defuzzification is carried out to convert the fuzzy output set to a "crisp" number, i.e. a clear and concise number, contained in the base of the output set. Two of the more common techniques are the "center of area" and "maximum" methods. All these methods are detailed in section 4.5. The interface of defuzzification uses one of these methods to assign an exact value to the information given from the decision logic and thus proposes a possible configuration of the IT-platform.

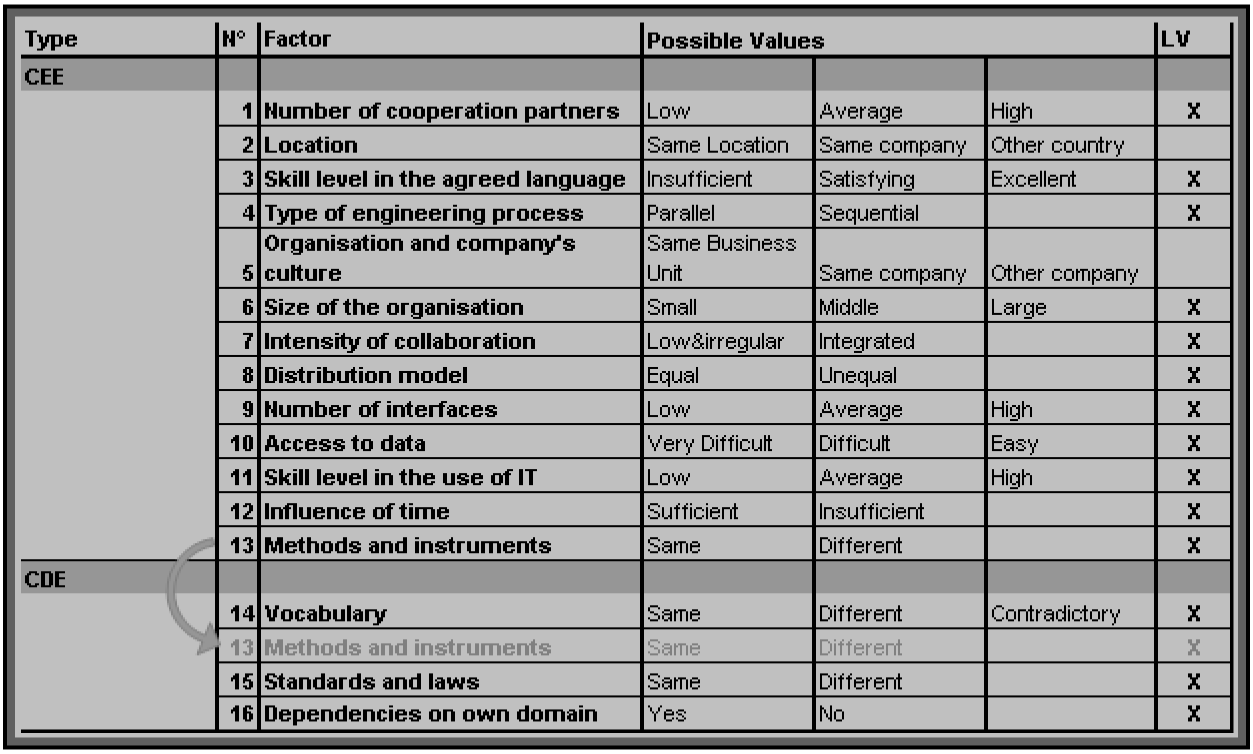
The defuzzification transforms the resulting output set in a "crisp" value, for example: this candidate is appropriate to 75%.

4.3. Fuzzification of the Influencing Factors

As already mentioned in section 4.2.4, the interface of fuzzification calculates the membership value from the input using the membership functions of each fuzzy set (see example in section 4.2.4). Toward this end, the influencing factors which have a fuzzy character must be defined as LR-Intervals or LR-Numbers. Each factor definition is justified separately with the help of previous analyses: [Richei, 1998], [Baumann et al., 2001], [Harms, 1995], and [ENSR, 2003]. For the factors ‘Distribution model’, ‘Number of interfaces’, ‘Vocabulary’, ‘Methods and instruments’, and ‘Standard and laws’ we could not find previous analyses that are detailed enough to enable the definition of LVs. Consequently, we defined them with the help of expert interviews within an industrial context. The description of the factors and the creation of the corresponding fuzzy sets are described in the following.

First, recall the influencing factors described in section 2.4 (see Table 4-1). As can be seen, they are not all linguistic variables (LV). In fact, the factors ‘Location’ and ‘Organisation and company’s culture’ are not vague and can be given by exact values. For example, the factor 5 ‘Organisation and company’s culture’ is not built as a fuzzy set, but rather as a singleton; in fact, it is possible to be in the same department, or in the same company, or in different companies, but not in both to a certain degree.

By defining the LV we have always chosen the parameters α and β , so that they are consistent (see section 4.2.3.2).



Type	N°	Factor	Possible Values			LV
CEE						
	1	Number of cooperation partners	Low	Average	High	X
	2	Location	Same Location	Same company	Other country	
	3	Skill level in the agreed language	Insufficient	Satisfying	Excellent	X
	4	Type of engineering process	Parallel	Sequential		X
	5	Organisation and company's culture	Same Business Unit	Same company	Other company	
	6	Size of the organisation	Small	Middle	Large	X
	7	Intensity of collaboration	Low&irregular	Integrated		X
	8	Distribution model	Equal	Unequal		X
	9	Number of interfaces	Low	Average	High	X
	10	Access to data	Very Difficult	Difficult	Easy	X
	11	Skill level in the use of IT	Low	Average	High	X
	12	Influence of time	Sufficient	Insufficient		X
	13	Methods and instruments	Same	Different		X
CDE						
	14	Vocabulary	Same	Different	Contradictory	X
	13	Methods and instruments	Same	Different		X
	15	Standards and laws	Same	Different		X
	16	Dependencies on own domain	Yes	No		X

Table 4-1: Influencing factors with fuzzy labels.

1. Number of cooperation partners

This characteristic shows how many partners are involved in the development process. Possible conditions of this characteristic are ‘low’, ‘average’, and ‘high’. A related problem is, for example, the flow of information. The information flow becomes more difficult with an increasing number of partners (cf. [Anderl et al., 1999b]).

Almost every study about cooperation projects distinguish between bilateral and multilateral cooperations. Subsequently, under a low number of cooperation partners, we understand a bilateral cooperation (cf. [Gierhardt, 2001], [Gierhardt et al. 1999], [Anderl et al. 1999a], [Anderl et al. 1999b], and [Baumann et al., 2001]). According to [Baumann et al., 2001], all cooperation projects having seven or more partners can be consider as very complex. From 3 to 6 partners, the number of cooperation partners is then considered as average. In this view, together with the definition of a consistent linguistic variable (see Definition 7 and Definition 8), the following linguistic variable results:

$$\begin{aligned}
&\textit{Linguistic Variable: Number of cooperation partners}\\
&T = \{Low, Average, High\}\\
&Base: [0,\infty)\textit{(partners)}\\
&\textit{Assignment of the term set to the base:}\\
&Low: \quad I_{1,L} = \langle 0,2,0,1 \rangle_{LR}\\
&Average: I_{1,A} = \langle 3,6,1,1 \rangle_{LR}\\
&High: \quad I_{1,H} = \langle 7,\infty,1,0 \rangle_{LR}
\end{aligned}$$

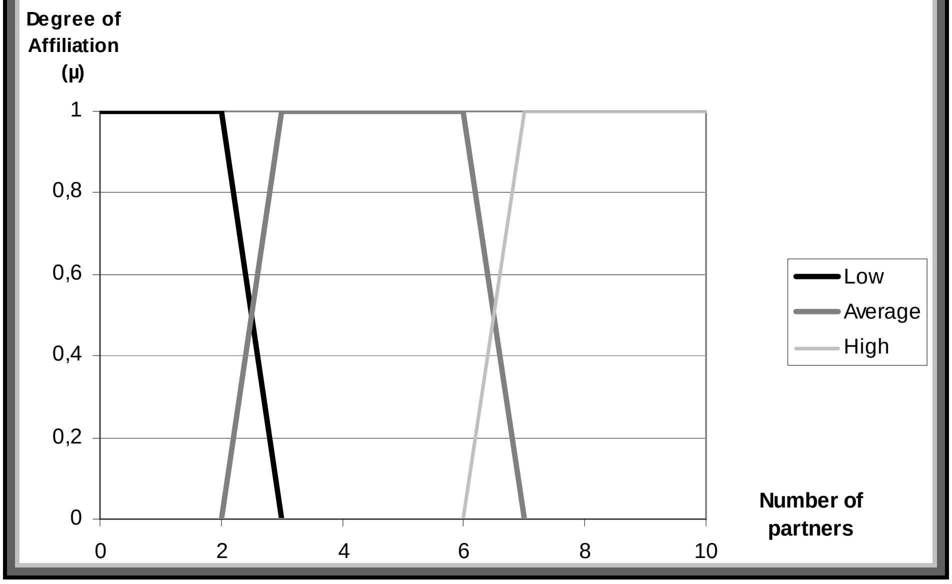


Figure 4-8: Representation of the linguistic variable 'Number of cooperation partners' (in accordance with [Gierhardt, 2001], [Gierhardt et al. 1999], [Anderl et al. 1999a], [Anderl et al. 1999b], and [Baumann et al., 2001]).

2. Location

The location of the cooperation partners also influences the project. In fact, the coordination between two partners becomes much more complicated when the distance is increased. ‘Same location, different room’; ‘same company, same country, different site’ or ‘different country’, are possible conditions of the characteristic ‘Location’. They describe where the partners are located during a distributed product development process. A typical problem for the condition ‘different country’ is the time change, if the various countries are in different time zones (cf. [Anderl et al., 1999b]). Another problem is the overcoming of cultural differences. For example, the organisation’s culture varies at the level of different units within the same site to the culture of the countries implicated in the project, like described in paragraphs 2.1.3 and 2.3. This factor is not defined as an LV.

3. Skill level in the agreed language

Usually in cooperation projects, in which the participants have different mother tongues, they agree on a common project language. Often English is chosen, and in fact it is the most widely taught and understood language in the world with approximately 1 to 1.5 billion speakers [Crystal, 1997]. However, no one can tell how well the project’s partners will be able to communicate in English and understand each other, which is also a problem for the project. This factor refers to the level of comprehension in a commonly used spoken language rather than to a discipline-specific language. It has also to be taken into account that the spoken as well as the written skill level influence team work.

This influencing factor has been defined as an LV in order to be an input of the fuzzy-expert-system, more precisely as a linguistic variable. The factor ‘Skill level in the agreed language’ is represented by a linguistic variable which has been defined by [Richei, 1998] and has the following characteristics:

Linguistic Variable : Skill level in the agreed language

$T = \{Unsufficient, Sufficient, Satisfying, Good, Excellent\}$

Base: 0 - 100 *(%)*

Assignment of the term set to the base :

$Unsufficient: \; I_{3,U} = \langle 0,66,0,8 \rangle_{LR}$

$Sufficient: \; I_{3,Su} = \langle 74,74,8,8 \rangle_{LR}$

$Satisfying: \; I_{3,Sa} = \langle 82,82,8,8 \rangle_{LR}$

$Good: \; I_{3,G} = \langle 90,90,8,8 \rangle_{LR}$

$Excellent: \; I_{3,E} = \langle 98,100,8,0 \rangle_{LR}$

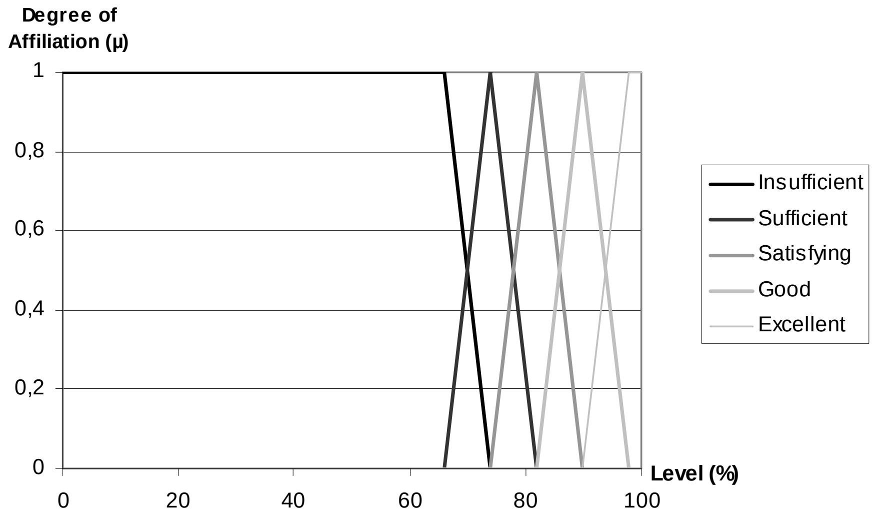


Figure 4-9: Representation of the linguistic variable 'Skill level in the agreed language' (in accordance with [Richei, 1998]).

4. Type of engineering process

The factor 'Type of engineering process' corresponds to the temporal relation between the different tasks from the focused part of the development process. These tasks may run in a sequential or in a parallel way. In practice a mixed form of the two types of engineering processes is observed. The synchronisation of different processes often arises as a problem for distributed projects. (cf. [Anderl et al., 1999b])

This influencing factor has also been defined as an LV. According to [Harms, 2005], the majority of the projects which are considered as conventional or following a sequential engineering process work simultaneously on 2 of 5 tasks. Generally parallel engineering processes work simultaneously on 4 of 5 tasks. In this view, together with the definition of consistent linguistic variables (see Definition 7 and Definition 8), the following variable results:

$$\begin{aligned}
&\textit{Linguistic Variable: Type of engineering process}\\
&T=\{Sequential, Parallel\}\\
&\textit{Base: 0 - 5 (number of tasks in parallel)}\\
&\textit{Assignment of the term set to the base:}\\
&Sequential:\ I_{4,S}=\langle 0,2,0,2\rangle_{LR}\\
&Parallel:\quad I_{4,P}=\langle 4,5,2,0\rangle_{LR}
\end{aligned}$$

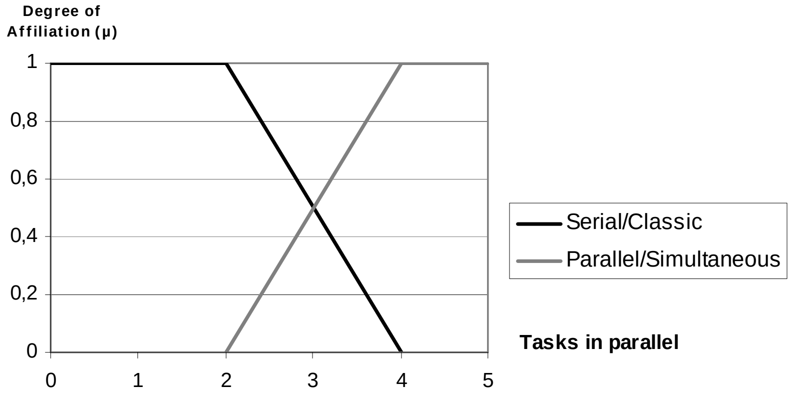


Figure 4-10: Representation of the linguistic variable 'Type of engineering process' (in accordance with [Harms, 1995]).

5. **Organisation and companies' culture**

The partners of the development process under consideration could belong to the same organisational unit, to the same company or to different companies. A problem occurring with partners of different companies is, for example, the gap between the cultures of the companies. In fact, they may have other habits, react differently to problems, and have other policies, strategies, and goals, which make the collaboration more difficult. This factor is not defined as an LV.

6. **Size of the organisation**

This characteristic describes the size of the related companies. The companies are classified into large scale enterprises, middle scale enterprises and small scale enterprises. On the one hand, a small-scale enterprise lacks an infrastructure. On the other hand, a large-scale enterprise may not be as flexible as a small one.

This influencing factor has also been defined as an LV. Officially, micro-enterprises have a maximum of 9 employees, small enterprises range from 10 to 49 employees and middle-scale companies consist of 50 to 249 employees. However, can we really use these numbers as a strict limit? If we remove one person from the employees of a large-scale enterprise, the enterprise remains large. Reapplying the concept over and over will ultimately lead to an enterprise with just one person. When does a large-scale enterprise become a middle-scale enterprise and then a small one? On the basis of statistical reports of the

European Commission, we have defined a linguistic variable corresponding to the factor. According to [ENSR, 2003] the average number of employees in a small enterprise is 19 persons; in middle-scale enterprises 98 employees and in big ones 1020 employees. In this view, together with the definition of consistent linguistic variables (see Definition 7 and Definition 8), the following variable results:

$$
\begin{aligned}
&\textit{Linguistic Variable: Size of the organisation}\\
&T = \{Small, Middle, Large\}\\
&Base: [0,\infty)(employees)\\
&\textit{Assignment of the term set to the base:}\\
&Small: \quad I_{6,S} = \langle 0,19,0,79 \rangle_{LR}\\
&Middle: I_{6,M} = \langle 98,98,79,922 \rangle_{LR}\\
&Large: \quad I_{6,L} = \langle 1020,\infty,922,0 \rangle_{LR}
\end{aligned}
$$

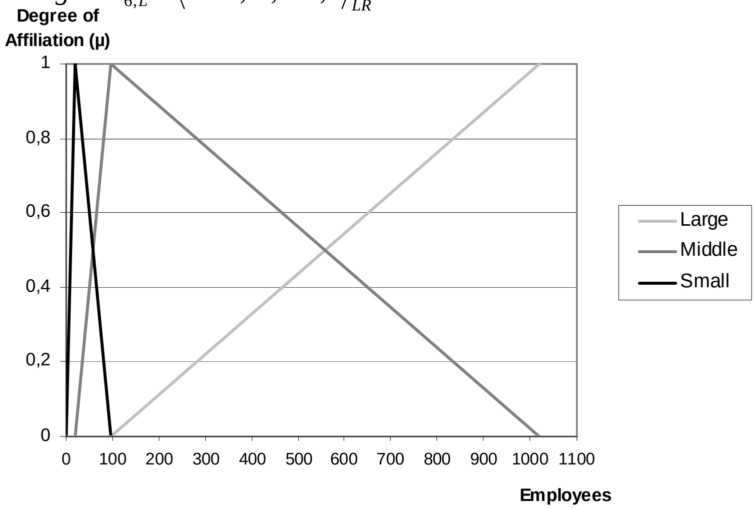


Figure 4-11: Representation of the linguistic variable 'Size of the organisation' (in accordance with [ENSR, 2003]).

7. Intensity of collaboration

The factor 'Intensity of collaboration' between the partners might be described as 'integrated' or 'a loose combination'. A possible problem that occurs with a low and irregular collaboration is the absence of a view on the global project: who is who, what do I have to do, etc. (cf. [Anderl et al., 1999b]). The collaboration takes place differently in both cases, namely, depending on whether the partners exchange only a few results or the collaboration is integrated into the daily work.

This influencing factor has also been defined as an LV. According to [Richei, 1998] the intensity of collaboration is represented by a linguistic variable which has the following characteristics:

$Linguistic\ Variable:\ Intensity\ of\ collaboration$

$T = \{VeryLow, Low, High, VeryHigh\}$

$Base:\ 0\text{ - }100\ (\%\ of\ daily\ work)$

$Assignment\ of\ the\ term\ set\ to\ the\ base:$

$VeryLow:\ I_{7,VL} = \langle 0,20,0,15 \rangle_{LR}$

$Low:\ I_{7,L} = \langle 35,35,15,12 \rangle_{LR}$

$High:\ I_{7,H} = \langle 47,47,12,15 \rangle_{LR}$

$VeryHigh:\ I_{7,VH} = \langle 62,100,15,0 \rangle_{LR}$

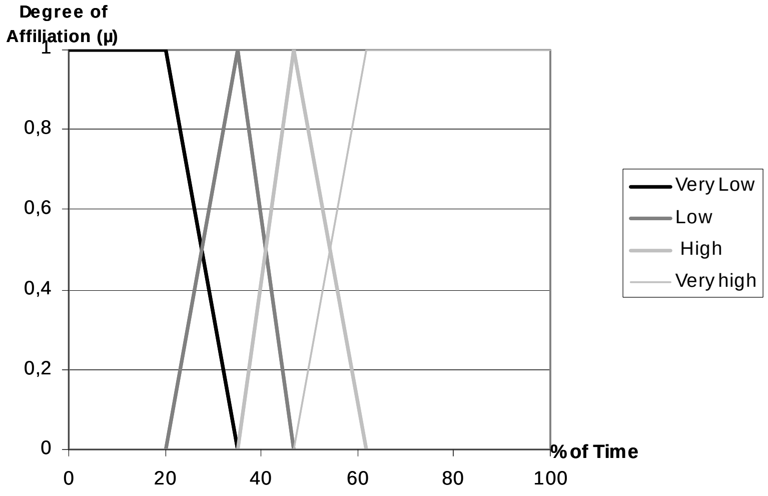


Figure 4-12: Representation of the linguistic variable 'Intensity of collaboration' (in accordance with [Richei, 1998]).

8. Distribution Model

The distribution of the tasks differs from one project to another and influences the cooperation processes. In fact, these tasks may not be equally distributed between the partners. For example, one partner can have the entire responsibility, whereas the other one only develops some parts of the product or software. This also influences the collaboration processes and the way the partners interact with each other's work.

So far there is no literature that permits defining an LV for this factor. Consequently, we decided to determine this fuzzy variable empirically. We asked the following questions to people who have experience in cooperation projects:

- In a cooperation project: when do you consider the tasks to be equally distributed between the partners?
- In a cooperation project: when do you consider the tasks to be unequally distributed between the partners?

The answers have all been given in the following form: Partner A: "… %", Partner B: …%.

The collected data are available in appendix 8.1 and can be considered as relevant since they verify the '68-95-99.7 rule' or 'Empirical Rule', which states that a normal distribution, also known as a bell curve or Gaussian distribution, is justified when about 68% of the values are within 1 standard deviation (σ) away from the mean (µ), about 95% of the values are within two standard deviations and about 99.7% lie within 3 standard deviations (cf. [Kenney et al., 1962], see Figure 4-13).

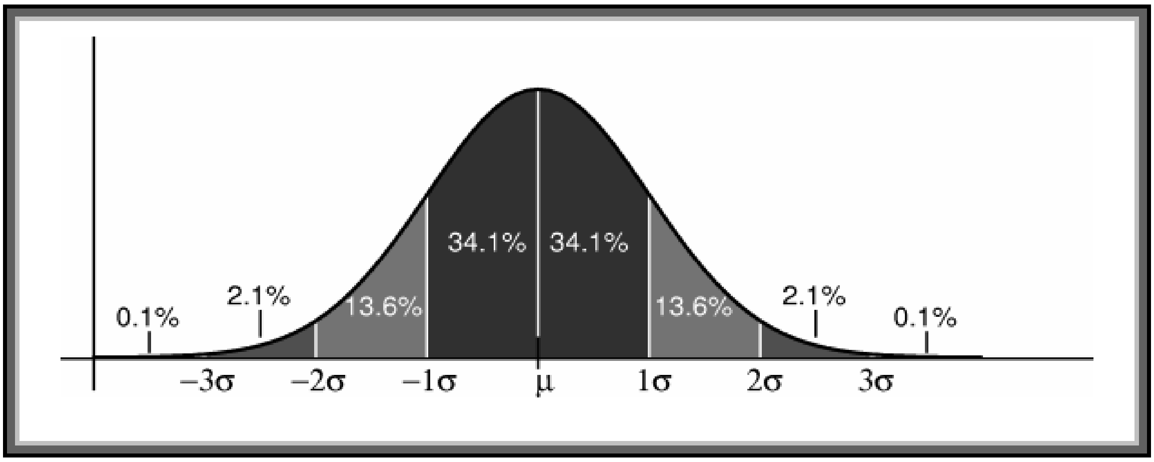


Figure 4-13: Normal distribution.

We calculate the difference $\Delta\%$ between the percentages of tasks carried out by the partners. This $\Delta\%$ is used to determine the peak of the fuzzy sets of this LV. This value $\Delta\%$ for each linguistic term 'Equally distributed' and 'Unequally distributed', together with the definition of a consistent LV (see Definition 7 and Definition 8), are sufficient to define a unique LV, which is:

$$\begin{array}{ll}
\textit{Linguistic Variable: Distribution model} & \\
T = \{EquallyDistributed, UnequallyDistributed\} & \\
\textit{Base}:\ 0\text{ - }100\ (\Delta\%) & \\
\textit{Assignment of the term set to the base:} & \\
\textit{Equally Distributed}: & I_{8,Ed} = \langle 0,18,0,20 \rangle_{LR} \\
\textit{Unequally Distributed}: & I_{8,Ud} = \langle 38,100,20,0 \rangle_{LR}
\end{array}$$

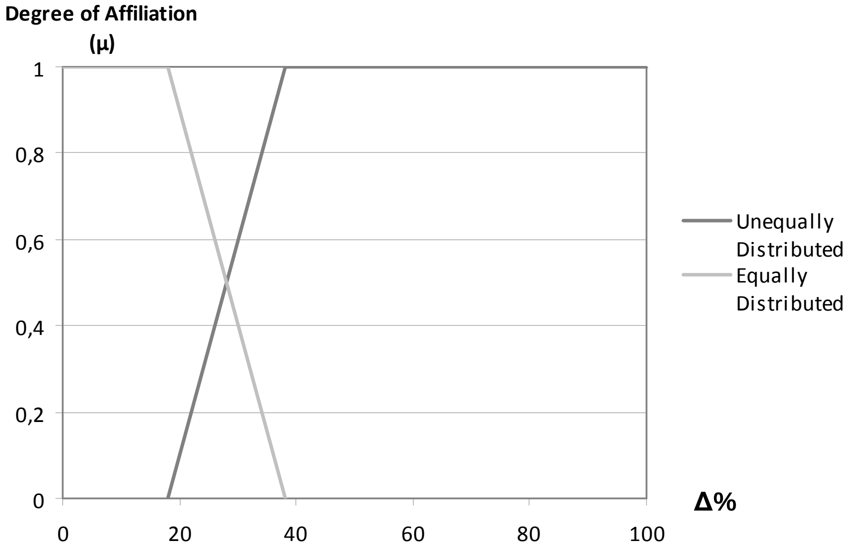


Figure 4-14: Representation of the linguistic variable ‘Distribution model’.

9. Number of Interfaces

The number of interfaces between all partners also increases the complexity, thus influencing the cooperation in an engineering project. The more technical and organisational interfaces one has, the more difficult it will be to transfer information clearly and completely.

So far there is no notice of literature that permits creating a fuzzy set of this factor. Subsequently, we decided to determine this LV empirically. We asked the following questions to people who have experience in cooperation projects:

- How many interfaces (technical or organisational) do you consider to be a low quantity of interfaces?
- How many interfaces (technical or organisational) do you consider to be an average quantity of interfaces?
- How many interfaces (technical or organisational) do you consider to be a high quantity of interfaces?
- How many interfaces (technical or organisational) do you consider to be a maximum?

The answers have all been given in the following form: "… interfaces", and correspond to the average number of interfaces for each term 'Low', 'Average', and 'High'.

The collected data are available in appendix 8.1 and can be considered as relevant because they verify the '68-95-99.7 rule' or 'Empirical Rule' already explained in the previous paragraph.

The collected values are used to set the peak of each fuzzy set. These values, together with the definition of a consistent LV (see Definition 7 and Definition 8), are sufficient to define a unique LV, which is:

$Linguistic\ \ Variable:\ \ Interfaces$

$T = \{Low, Average, High\}$

$Base:\ 0 - \infty (interfaces)$

$Assignment\ of\ \ the\ term\ set\ to\ the\ base:$

$Low: \qquad I_{9,L} = \langle 0,3,0,4 \rangle_{LR}$

$Average:\ I_{9,A} = \langle 7,7,4,6 \rangle_{LR}$

$High: \qquad I_{9,H} = \langle 13,\infty,6,0 \rangle_{LR}$

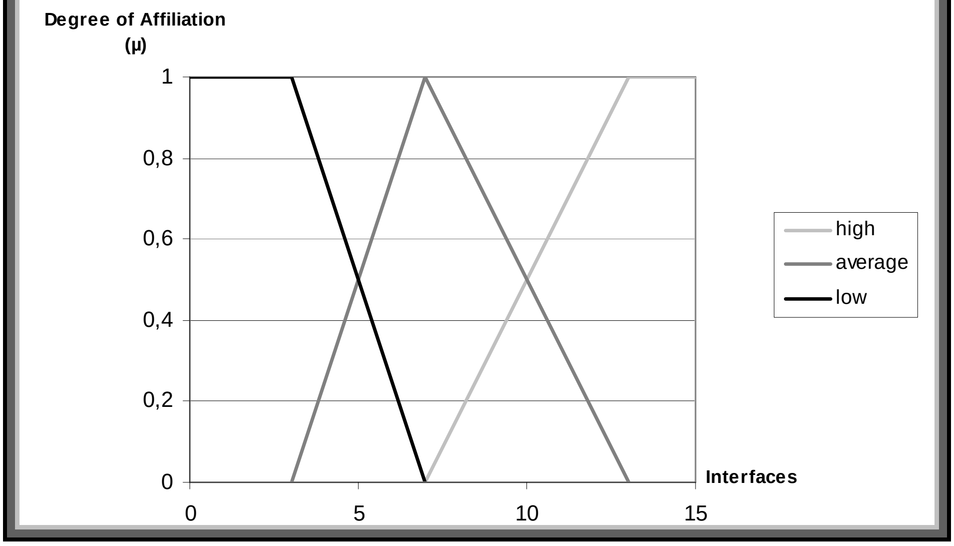


Figure 4-15: Representation of the linguistic variable 'Interfaces'.

10. Access to data

This point characterizes the technical and organisational ability of every partner to access the data that are relevant for an engineering project (e.g. access authorization, existing infrastructures,...). The 'Access to data' might be 'very difficult', 'difficult' or 'easy'. On the one hand, a difficult access to data, for example through a firewall, reduced access authorisations or secrecy rules, increases the complexity of the project. On the other hand, if access to all data for every partner is possible, the consistency of the data might be a problem (cf. [Anderl et al. 1999b]).

This influencing factor has also been defined as an LV. According to [Richei, 1998] the access to data is represented by a linguistic variable which has the following characteristics:

$Linguistic\ Variable:\ Access\ to\ data$

$T = \{VeryDifficult, Difficult, Easy\}$

$Base:\ 0\text{ - }5$

$Assignment\ of\ the\ term\ set\ to\ the\ base:$

$Very\ Difficult:\ I_{10,Vd} = \langle 0,2,0,1 \rangle_{LR}$

$Difficult:\ I_{10,D} = \langle 3,3,1,1 \rangle_{LR}$

$Easy:\ I_{10,E} = \langle 4,5,1,0 \rangle_{LR}$

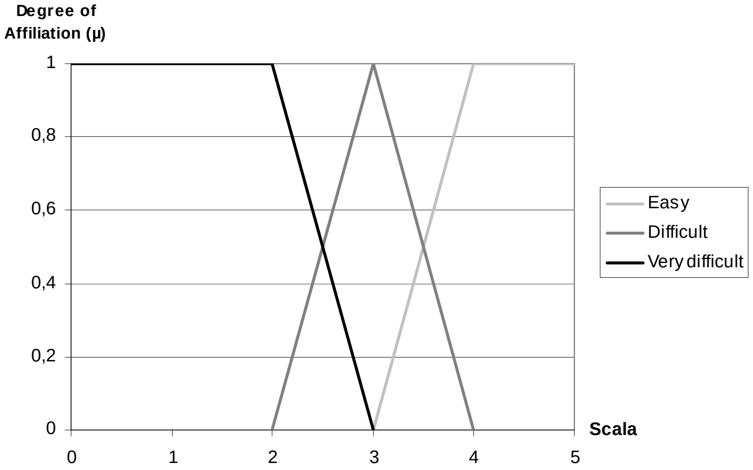


Figure 4-16: Representation of the linguistic variable 'Access to data' (in accordance with [Richei, 1998]).

11. Skill level in the use of IT

This factor describes the global competence of the project participants in the use of IT. We focused on the original factor 'Competence' of [Gierhardt et al. 1999], [Anderl et al. 1999a] and [Anderl et al. 1999b] on the skill level in the use of IT because our goal is to configure an optimal IT-platform corresponding to given projects. The level of the participants might be low, average or high. For example, if the competence is low, the platforms proposed should be basic and intuitive.

This influencing factor has also been defined as an LV. According to [Richei, 1998] the skill level in the use of IT is represented by a linguistic variable which has the following characteristics:

$Linguistic\ \ Variable:\ Skill\ level\ in\ the\ use\ of\ \ IT$

$T = \{Low, Average, High\}$

$Base:\ 0 - 10$

$Assignment\ of\ \ the\ term\ set\ to\ the\ base:$

$Low: \quad I_{11,L} = \langle 0,2,0,2 \rangle_{LR}$

$Average: \quad I_{11,A} = \langle 4,7,2,2 \rangle_{LR}$

$High: \quad I_{11,H} = \langle 9,10,2,0 \rangle_{LR}$

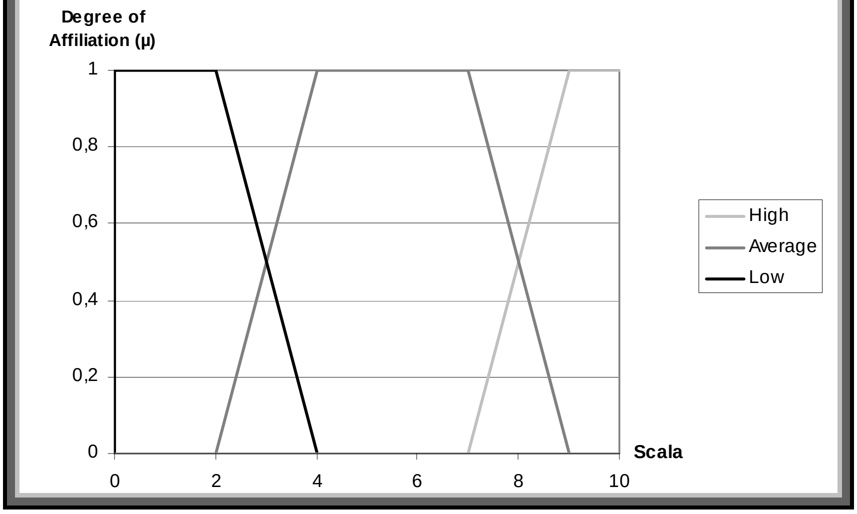


Figure 4-17: Representation of the linguistic variable 'Skill level in the use of IT' (in accordance with [Richei, 1998]).

12. Influence of time

This factor corresponds to the factor 'Capacity' of [Gierhardt et al. 1999], [Anderl et al. 1999a] and [Anderl et al. 1999b]. They define it as the available time to finalise the project (also cf. [Gaul, 2001]). The possible conditions are sufficient or insufficient. The less time one has to finish a project, the less time one will have to adjust your collaboration. That is why the remaining available time for a project influences the quality of collaboration.

This influencing factor has also been defined as a fuzzy set. According to [Richei, 1998] the influence of time is represented by a linguistic variable which has the following term set $T = \{Satisfactory, Sufficient, Critical, Transgressed\}$ and the characteristics:

$Linguistic\ \ Variable:\ \ Influence\ of\ \ time$

$T = \{Satisfactory, Sufficient, Critical, Transgressed\}$

$Base:\ 0\text{ - }100\,(\%)$

$Assignment\ of\ \ the\ term\ set\ to\ the\ base:$

$Satisfactory:\ \ I_{12,Sa} = \langle 0,40,0,10 \rangle_{LR}$

$Sufficient:\ \ \ \ I_{12,Su} = \langle 50,60,10,10 \rangle_{LR}$

$Critical:\ \ \ \ \ \ I_{12,C} = \langle 70,80,10,20 \rangle_{LR}$

$Transgressed:\ I_{12,T} = \langle 100,100,20,0 \rangle_{LR}$

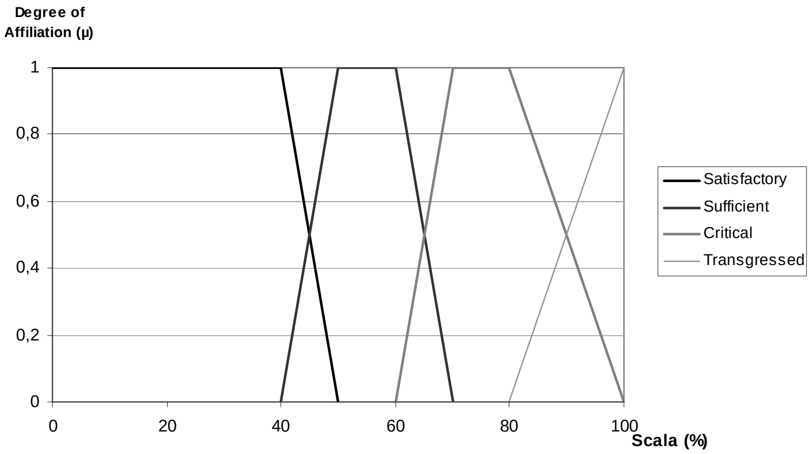


Figure 4-18: Representation of the linguistic variable 'Influence of time' (in accordance with [Richei, 1998]).

13. Methods and instruments

The factor 'Methods and Instruments' covers the compatibility of the participants' instruments and tools as well as the compatibility of the partners' methods. This factor is relevant to characterize the distribution of the projects under consideration together with its' interdisciplinarity.

In fact, the methods and instruments employed not only depend on the participants' organisations and distribution, but also on their specialisation, background and fields of work (see section 2.3 and Figure 2-3). If the members of a project are specialists in various disciplines, they may use different methods and instruments and follow different processes that may influence the communication within the project. If tools or instruments (e.g. CAD-system) are not compatible, data exchange will be a problem. If there is no compatibility of methods the coordination of working together will be difficult (cf. [Anderl et al. 1999b]).

Until now no literature exists that permits defining an LV of this factor. Consequently, we decided to determine this LV empirically. In accordance with [Gierhardt, 2001], [Gierhardt et al. 1999], [Anderl et al. 1999a], and [Anderl et al. 1999b], we chose to remain with only the two terms 'Same' and 'Different'. If possible, it is an advantage to keep the LVs simple with few terms, since each term requires new rules in the rule base and therefore a longer and more complex empiricism. We didn’t want to bother our experts with additional questions and so tried to define the LVs as simple as possible unless otherwise noted. We asked the following questions to people who have experience in cooperation projects:

- In a development project the participants may use different methods and instruments. If 100% means that the methods and instruments are totally equal: at which percentage would you consider the methods and instruments employed as the same or equivalent?
- In a development project the participants may use different methods and instruments. If 100% means that the methods and instruments are totally equal: at which percentage would you consider the methods and instruments employed as different?

The answers have all been given in the following form: “… %” and subsequently represent the maximum of the two fuzzy sets ‘Different’ and ‘Same’.

The collected data are available in appendix 8.1 and are considered to be relevant because they verify the ‘68-95-99.7 rule’ or ‘Empirical Rule’ already explained in a previous paragraph. The following LV results then from the collected data and the definition of a consistent LV (see Definition 7 and Definition 8):

Linguistic Variable : Methods and instruments

$T = \{Different, Same\}$

Base: *0 - 100 (%)*

Assignment of the term set to the base :

Different : $I_{13,D} = \langle 0,56,0,27 \rangle_{LR}$

Same : $I_{13,S} = \langle 83,100,27,0 \rangle_{LR}$

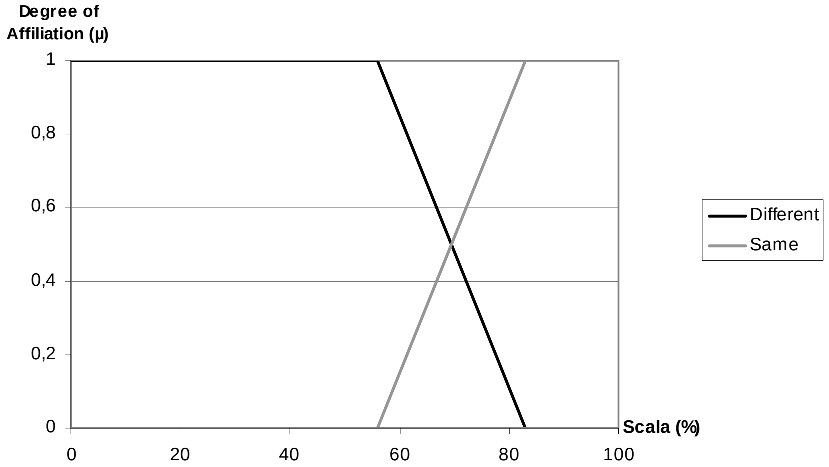


Figure 4-19: Representation of the linguistic variable ‘Methods and Instruments’.

At this point it is interesting to note that methods, tools or instruments that correlate to 83% are considered by the interviewed persons as the same or equivalent. Such a consideration points forward new opportunities to reduce development costs. In fact, taking it into account could avoid the development of methods and tools that only differ slightly from existing ones thus producing unnecessary costs.

14. Vocabulary

The specialists involved have their own vocabulary, not only depending on their organisation, but also on their domain, background and knowledge. The participants of the project could be specialists in various disciplines. Therefore, they may use different terms, expressions or abbreviations; in the worst case, the same terms can have different meanings and cause contradictions. The factor ‘Vocabulary’ might be described as ‘Same’ when the participants use the same terms with the same meanings, as ‘Different’ when they have different vocabularies from their original discipline, or as ‘Contradictory’ when the vocabulary leads to contradiction and incomprehension. This often happens, by the existence of superficial similarities, for instance identical words used with quite different meanings. All this may cause difficulties of understanding.

So far no literature exists that permits creating an LV of this factor. Subsequently, we decided to determine this LV empirically. In accordance with [Gierhardt, 2001], [Gierhardt et al. 1999], [Anderl et al. 1999a], and [Anderl et al. 1999b], we chose to remain with only the three terms ‘Same’, ‘Different’, and ‘Contradictory’. If possible, it is an advantage to keep the LVs simple with few terms, since each term requires new rules in the rule base and therefore a longer and more complex empiricism. We didn’t want to bother our experts with additional questions and so tried to define the LVs as simple as possible unless otherwise noted. We asked the following questions to people who have experience in cooperation projects:

- In a development project the participants may use different terms, expressions or abbreviations specific to their discipline. If 100% means that the participants use the exact same vocabulary: at which percentage would you consider the vocabulary employed as same or equivalent?
- In a development project the participants may use different terms, expressions or abbreviations specific to their discipline. If 100% means that the participants use the exact same vocabulary: at which percentage would you consider the vocabulary employed as different?
- In a development project the participants may use different terms, expressions or abbreviations specific to their discipline. If 100% means that the participants use the exact same vocabulary: at which percentage would you consider the vocabulary employed as contradictory?

The answers have all been given in the following form: “… %” and subsequently represent the maximum of the three fuzzy sets ‘Different’, ‘Same’, and ‘Contradictory’.

The collected data are available in appendix 8.1 and are considered to be relevant since they verify the ‘68-95-99.7 rule’ or ‘Empirical Rule’ already explained in a previous paragraph. The following LV results then from the collected data and the definition of a consistent LV (see Definition 7 and Definition 8):

$$Linguistic\ Variable: Vocabulary$$

$$T = \{Contradictory, Different, Same\}$$

$$Base: 0 - 100\,(\%)$$

$$Assignment\ of\ the\ term\ set\ to\ the\ base:$$

$$Contradictory: \quad I_{14,C} = \langle 0,22,0,26 \rangle_{LR}$$

$$Different: \quad I_{14,D} = \langle 48,48,26,25 \rangle_{LR}$$

$$Same: \quad I_{14,S} = \langle 73,100,25,0 \rangle_{LR}$$

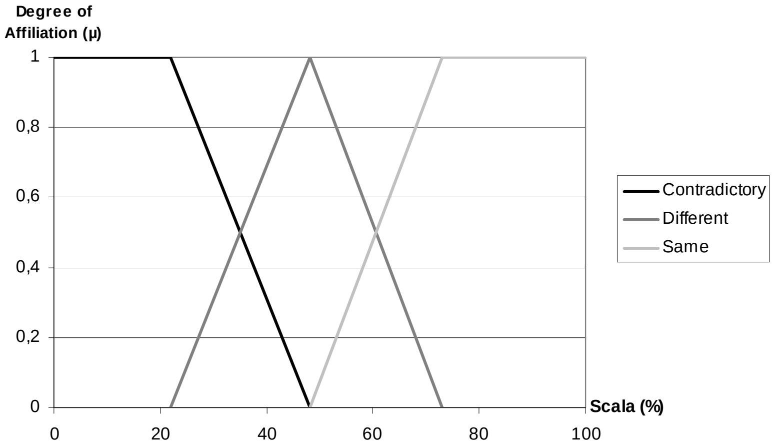


Figure 4-20: Representation of the linguistic variable 'Vocabulary'.

15. Standards and laws

'Standards and laws' is principally a factor of cross-domain contexts, which can have as possible conditions: same or different. If the people in a project are specialists in various disciplines, they may have different standards, laws, references, etc. relative to their field of work which may influence the communication within the project (see section 2.3 and Figure 2-3).

So far no literature exists that permits creating an LV of this factor. Consequently, we decided to determine this LV empirically. In accordance with [Gierhardt, 2001], [Gierhardt et al. 1999], [Anderl et al. 1999a], and [Anderl et al. 1999b], we chose to remain with only the two terms 'Same' and 'Different'. If possible, it is an advantage to keep the LVs simple with few terms, since each term requires new rules in the rule base and therefore a longer and more complex empiricism. We didn't want to bother our experts with additional questions and so tried to define the LVs as simple as possible unless otherwise noted. We asked the following questions to people who have experience in cooperation projects:

- In a development project the participants may have different standards, laws, references, etc. relative to their field of work. If 100% means that the standards and laws are totally identical: at which percentage would you consider the standards and laws followed as the same or equivalent?

- In a development project the participants may have different standards, laws, references, etc. relative to their field of work. If 100% means that the standards and laws are totally identical: at which percentage would you consider the standards and laws followed as different?

The answers have all been given in the following form: "… %" and subsequently represent the maximum of the two fuzzy sets 'Different' and 'Same'.

The collected data are available in appendix 8.1 and are considered to be relevant because they verify the '68-95-99.7 rule' or 'Empirical Rule' already explained in a previous paragraph. The following LV results then from the collected data and the definition of a consistent LV (see Definition 7 and Definition 8):

$$\begin{aligned}
&\textit{Linguistic Variable: Standards and laws}\\
&T = \{\textit{Different}, \textit{Same}\}\\
&\textit{Base}:\ \textit{0 - 100 (\%)}\\
&\textit{Assignment of the term set to the base:}\\
&\textit{Different}: \quad I_{15,D} = \langle 0,69,0,21 \rangle_{LR}\\
&\textit{Same}: \quad I_{15,S} = \langle 90,100,21,0 \rangle_{LR}
\end{aligned}$$

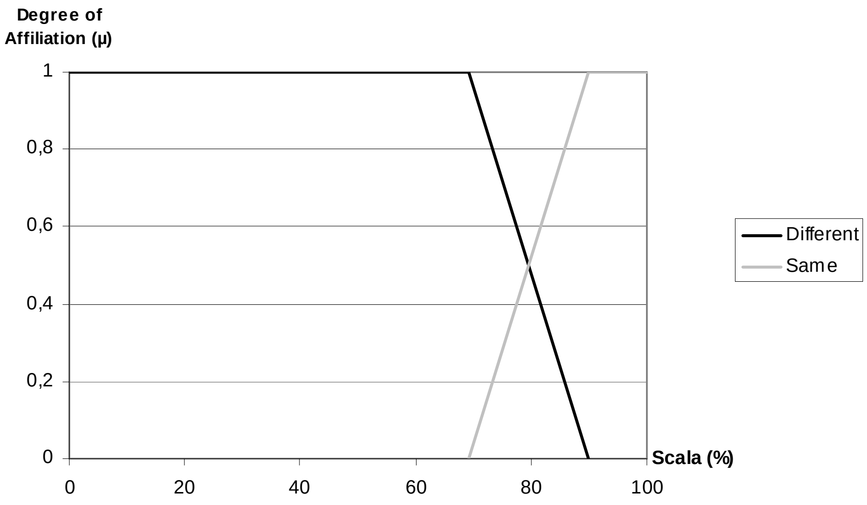


Figure 4-21: Representation of the linguistic variable 'Standards and laws'.

16. Dependencies

The factor 'Dependencies' has already been commented on in section 2.3. If the people in a project are specialists in various disciplines, they may have their own discipline paradigm. They depend on a community and may have prejudices against other disciplines. If they depend too much on a community, group or organisation (e.g. a scientific community, its organisation, a development team, etc.), they will probably lack the required flexibility to be a part of such a cross-domain project. Thus, the factor 'Dependencies' influences the project's cooperation and can have as possible conditions: influenced by a community or not influenced.

This influencing factor has also been defined as an LV. According to [Richei, 1998] the factor 'Dependencies' is represented by a linguistic variable having the following term set $T = \{No, Lightly, Middle, High, Totally\}$ and the characteristics:

$Linguistic\ \ Variable:\ \ Dependencies$

$T = \{No, Lightly, Middle, High, Totally\}$

$Base: [0,1]$

$Assignment\ of\ \ the\ term\ set\ to\ the\ base:$

$No: \qquad I_{16,N} = \langle 0,0,0,0.05 \rangle_{LR}$

$Lightly: \quad I_{16,L} = \langle 0.05,0.05,0.05,0.09 \rangle_{LR}$

$Middle: \quad I_{16,M} = \langle 0.14,0.14,0.09,0.36 \rangle_{LR}$

$High: \qquad I_{16,H} = \langle 0.5,0.5,0.36,0.5 \rangle_{LR}$

$Totally: \quad I_{16,T} = \langle 1,1,0.5,0 \rangle_{LR}$

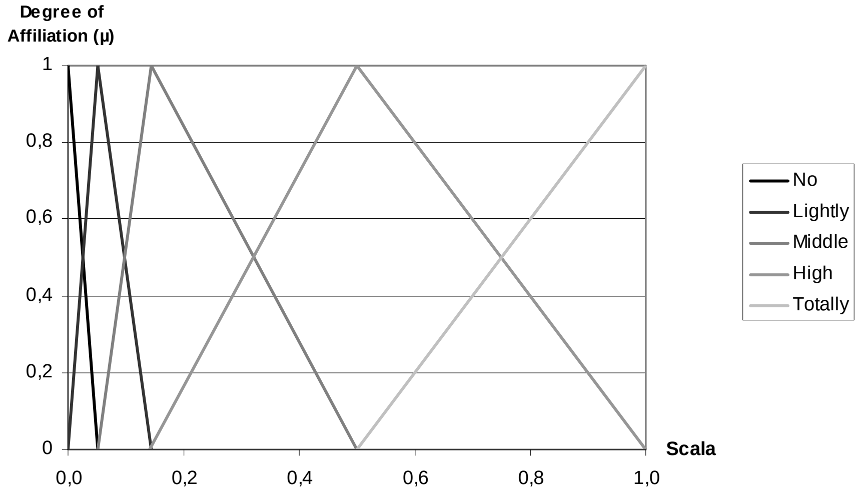


Figure 4-22: Representation of the linguistic variable 'Dependencies' (in accordance with [Richei, 1998]).

4.4. The Inference Process

As mentioned in section 4.1 knowledge-based systems are computer programs that contain subject-specific knowledge of one or more human experts. The knowledge base an expert uses is what he learned at school, from colleagues, and from years of experience. Presumably, the more experience he has, the larger his store of knowledge. Knowledge allows an expert to interpret the information in his own databases and thus take advantages in diagnosis, design, and analysis. The problem-solving model organizes this knowledge with the help of rules to solve the problem.

4.4.1. The Knowledge-Base

An empirical study has been conducted to create the earlier-mentioned base of knowledge of the expert-system's decision logic. In this study the IT-tools supporting collaboration and communication which are candidates to becoming part of the collaboration platform have been rated by experts in the support of cooperation projects regarding the various influencing factors in order to build the rules of our fuzzy-expert system. In this section we comment the global results of the empirical study and in addition, give some advice about the processes concerned. At this point, it should be remembered that the fuzzy-expert-system deals with standard tools and collaboration processes as described in chapter 3.2.

As reported in [Laurent, 2006] and [Laurent, 2007a] this empirical research shows that the preferred tools are electronic calendars, project management systems, document management systems, workflow systems, and E-mail systems. In addition, it turned out that the majority of the users would like to use IT-support-systems also to organize and coordinate activities.

Electronic calendars, project management systems and workflow systems are typical systems to support coordination processes, like, for example, organising reviews and meetings, planning, tasks assignment, definition of tasks, etc. In principal it handles the management of dependencies among tasks and agents in an enterprise in order to reduce time and costs (cf. [Petrie, 1998]. Electronic calendars offer the possibility to schedule events and be notified and reminded automatically about them. They are also often used as personal or group tasks list. A project management system offers more by enabling also the planning of objectives and resources, acquiring material and human resources, tracking and reporting progress, etc. (cf. [PMI, 2004]). A workflow system is a functionality which manages documents and their processing. A designer determines what kinds of actions need to be taken under what conditions by whom and encodes this into a workflow process. People assume roles in processing the documents (cf. [Petrie, 1998]).

E-Mail systems, as well as document management systems, support in principal communication and information processes. Nevertheless, E-Mail systems in contrast to document management systems seem to be inappropriate for big cooperation projects with participants speaking different languages. In fact, the use of E-Mails is often disorganised, chaotic and in some cases superfluous, which can lead to confusion in big cooperation projects, especially if the participants cannot express themselves well enough in the chosen project language. In this case document management systems are considered more appropriate, since they are often structured and offer the possibility to categorise documents, manage new versions, generate E-Mail-alerts in a structured way, etc. Regarding the project language, a document management system supports in most cases a higher quality of information while authoring the document than E-Mails. In addition, document management systems often have a multi-language support for the user interface, as well as content. However they require a higher skill level in the use of IT.

Communication and collaboration tools like online-conferencing or electronic meeting systems first aim at reproducing virtually the conditions of a meeting. This means that they support communication processes such as information exchange, discussions, decision making, agreements, etc. as well as collaborative processes such as application sharing, online-workshops, and collaborative development. Electronic meeting systems are often more appropriate for collaborative tasks, since they offer technically more possibilities, like electronic brainstorming, categorizer, or topic commenter. Nevertheless, online-conferencing, as well as electronic meeting systems, are often seen as too complex and make sense only in a non-cross-domain context according to the results. However, for small projects with participants speaking the same language and being in a familiar environment, these tools are often rated as very useful.

Instant-messaging and chat rooms have been rated as inappropriate for almost every project configuration. These tools support communication processes in an ad-hoc and informal way, as the phone does. Due to this character they seem not to be appropriate neither for decision making nor for formal information exchange, task assignments or coordination processes. They enable a synchronous communication, which is very complicated to manage for cooperation projects across different countries, companies, and time zones. Finding by chance the right person at the right place and at the right moment is considered too big a challenge. In addition, instant-messaging and chat rooms are often considered as too tedious because of the typing and latencies between responses, which is a disadvantage for an ad-hoc and informal synchronous communication medium, in comparison to the phone for example.

Whiteboards, forums, social software, co-authoring systems, and problem-solving systems find acceptance to some degree. In fact, users seem to be attracted by these tools, because they offer in part new opportunities, while fearing at the same time that no one will use them because they are not part of their daily work.

Forums support only discussions and information exchange, whereas the other tools are far more sophisticated.

Whiteboards are often complementary to an online-conferencing tool; in fact, they allow more than one person to work on an image or text at any given time, and thus serve also for demonstration and explanation purposes. In a most simple way a whiteboard can act as a simple information board.

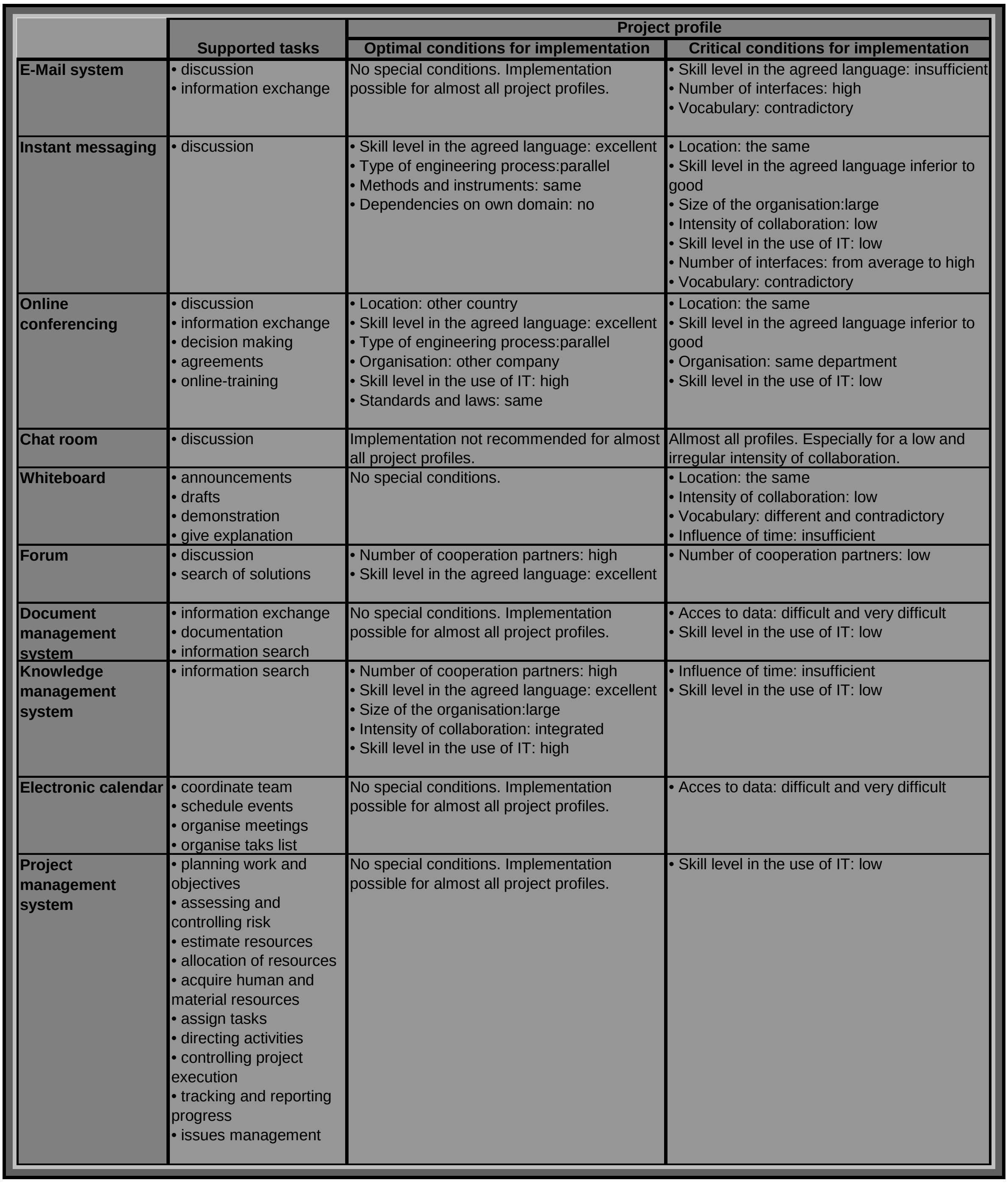
A social software system permits the creation of networks. It organises groups according to their field of work, competence or domains, so that they can connect together.

Co-authoring systems rather serve communication or collaboration processes than coordination processes. Such a system permits the spread of information or knowledge within an organisation, as well as a collaborative work on a document, task or topic. In this way it enables not only information exchange, but also a collaborative work or activity. Finally problem-solving systems support creative tasks and brainstorming by providing electronically common techniques of problem-solving such as lateral thinking puzzles or morphological analysis.

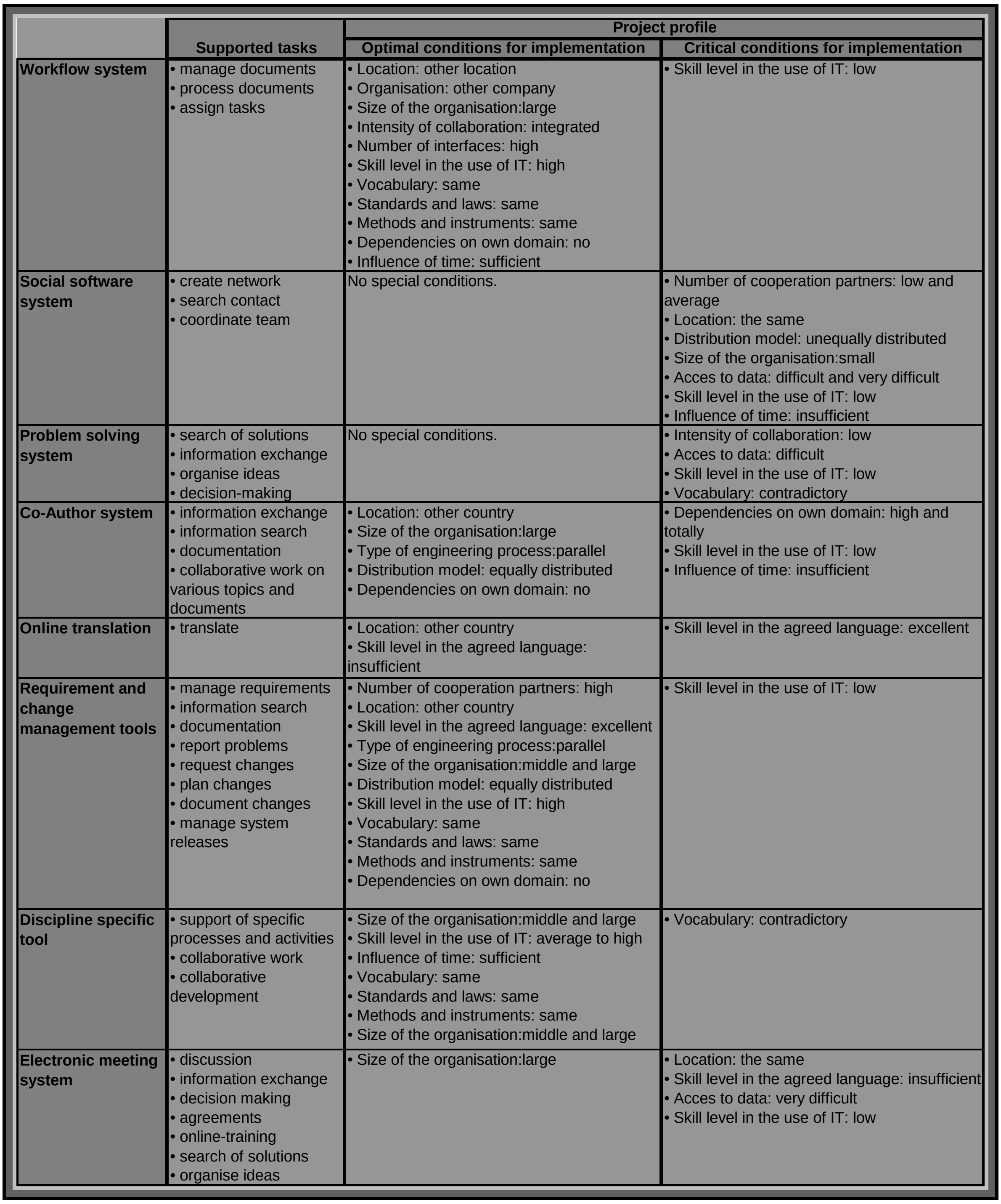
Knowledge management systems enable the spread and organisation of knowledge within an enterprise but are not the result of a collaborative work. They act as knowledge-base of an organisation or group. They are granted a middle acceptance, but are generally well received by participants in cooperation projects with important cultural differences or sensitive work contexts, as well as requirements and change management tools, which support exclusively requirements and change management processes in each kind of domain.

Discipline specific tools support discipline specific processes, as already mentioned in section 3.1.2. This category of tools can not be considered as standard, since they support only specific processes. Although discipline specific tools are applicable in general, they seem to be best suited at the collaboration level to small projects with participants from the same discipline. In fact, in a strong cross-domain context, it is inappropriate to make a discipline specific tool available on the collaboration-platform if one or two specialists only can use it.

The following table summarizes the tasks supported by each IT-tool and the corresponding project profiles. Since we do not base our work on processes or tasks but on the project configuration, the table does not present all possible applications of a tool, but gives some examples of the main ones. The results of the empirical study are the conditions of implementation of a tool, and can be found partly in the columns “Optimal conditions for implementation” and “Critical conditions for implementation” in the form “Influencing factor: value in the considered project” (see Table 4-2). This pair “<FACTOR>: <VALUE>” basically describes the project profile. The entire results of the empirical study enable the creation of fuzzy rules.



		Project profile	
	Supported tasks	**Optimal conditions for implementation**	**Critical conditions for implementation**
E-Mail system	• discussion • information exchange	No special conditions. Implementation possible for almost all project profiles.	• Skill level in the agreed language: insufficient • Number of interfaces: high • Vocabulary: contradictory
Instant messaging	• discussion	• Skill level in the agreed language: excellent • Type of engineering process:parallel • Methods and instruments: same • Dependencies on own domain: no	• Location: the same • Skill level in the agreed language inferior to good • Size of the organisation:large • Intensity of collaboration: low • Skill level in the use of IT: low • Number of interfaces: from average to high • Vocabulary: contradictory
Online conferencing	• discussion • information exchange • decision making • agreements • online-training	• Location: other country • Skill level in the agreed language: excellent • Type of engineering process:parallel • Organisation: other company • Skill level in the use of IT: high • Standards and laws: same	• Location: the same • Skill level in the agreed language inferior to good • Organisation: same department • Skill level in the use of IT: low
Chat room	• discussion	Implementation not recommended for almost all project profiles.	Allmost all profiles. Especially for a low and irregular intensity of collaboration.
Whiteboard	• announcements • drafts • demonstration • give explanation	No special conditions.	• Location: the same • Intensity of collaboration: low • Vocabulary: different and contradictory • Influence of time: insufficient
Forum	• discussion • search of solutions	• Number of cooperation partners: high • Skill level in the agreed language: excellent	• Number of cooperation partners: low
Document management system	• information exchange • documentation • information search	No special conditions. Implementation possible for almost all project profiles.	• Acces to data: difficult and very difficult • Skill level in the use of IT: low
Knowledge management system	• information search	• Number of cooperation partners: high • Skill level in the agreed language: excellent • Size of the organisation:large • Intensity of collaboration: integrated • Skill level in the use of IT: high	• Influence of time: insufficient • Skill level in the use of IT: low
Electronic calendar	• coordinate team • schedule events • organise meetings • organise taks list	No special conditions. Implementation possible for almost all project profiles.	• Acces to data: difficult and very difficult
Project management system	• planning work and objectives • assessing and controlling risk • estimate resources • allocation of resources • acquire human and material resources • assign tasks • directing activities • controlling project execution • tracking and reporting progress • issues management	No special conditions. Implementation possible for almost all project profiles.	• Skill level in the use of IT: low



	Supported tasks	Project profile	
		Optimal conditions for implementation	**Critical conditions for implementation**
Workflow system	• manage documents • process documents • assign tasks	• Location: other location • Organisation: other company • Size of the organisation:large • Intensity of collaboration: integrated • Number of interfaces: high • Skill level in the use of IT: high • Vocabulary: same • Standards and laws: same • Methods and instruments: same • Dependencies on own domain: no • Influence of time: sufficient	• Skill level in the use of IT: low
Social software system	• create network • search contact • coordinate team	No special conditions.	• Number of cooperation partners: low and average • Location: the same • Distribution model: unequally distributed • Size of the organisation:small • Acces to data: difficult and very difficult • Skill level in the use of IT: low • Influence of time: insufficient
Problem solving system	• search of solutions • information exchange • organise ideas • decision-making	No special conditions.	• Intensity of collaboration: low • Acces to data: difficult • Skill level in the use of IT: low • Vocabulary: contradictory
Co-Author system	• information exchange • information search • documentation • collaborative work on various topics and documents	• Location: other country • Size of the organisation:large • Type of engineering process:parallel • Distribution model: equally distributed • Dependencies on own domain: no	• Dependencies on own domain: high and totally • Skill level in the use of IT: low • Influence of time: insufficient
Online translation	• translate	• Location: other country • Skill level in the agreed language: insufficient	• Skill level in the agreed language: excellent
Requirement and change management tools	• manage requirements • information search • documentation • report problems • request changes • plan changes • document changes • manage system releases	• Number of cooperation partners: high • Location: other country • Skill level in the agreed language: excellent • Type of engineering process:parallel • Size of the organisation:middle and large • Distribution model: equally distributed • Skill level in the use of IT: high • Vocabulary: same • Standards and laws: same • Methods and instruments: same • Dependencies on own domain: no	• Skill level in the use of IT: low
Discipline specific tool	• support of specific processes and activities • collaborative work • collaborative development	• Size of the organisation:middle and large • Skill level in the use of IT: average to high • Influence of time: sufficient • Vocabulary: same • Standards and laws: same • Methods and instruments: same • Size of the organisation:middle and large	• Vocabulary: contradictory
Electronic meeting system	• discussion • information exchange • decision making • agreements • online-training • search of solutions • organise ideas	• Size of the organisation:large	• Location: the same • Skill level in the agreed language: insufficient • Acces to data: very difficult • Skill level in the use of IT: low

Table 4-2: Application and conditions of implementation of the IT-tools.

4.4.2. Sensitivity Analysis

One important point to be considered for the creation of the inference rules is how to weigh each influencing factor. In fact, a brief overview of the results of the empirical study shows that some influencing factors have much more impact on the choice of the experts than the others. Moreover, it is not always the case that the same factors have an impact on the choice of one tool or another. To evaluate the weight of each factor we have to quantify the dependence of the output on the input parameters. To this end a sensitivity analysis is executed.

In general, sensitivity analysis is the study of how the uncertainty in the output of a model can be attributed to different sources of uncertainty in the model input. Various questions can be answered by a sensitivity analysis. According to [Saltelli, 2004] it is important to specify the purpose of the analysis before starting it. We want to determine a ranking of the influencing factors. In this context one setting of sensitivity analysis is interesting: factor prioritisation.

In Factor Prioritisation (FP), the question addressed is: what factor, once fixed to its true albeit unknown value, would give the greatest reduction in the uncertainty of the output? This factor is then the most important factor. Likewise the second most important factor can be defined and so on (cf. [Saltelli et al., 2000] and [Saltelli et al., 2004]).

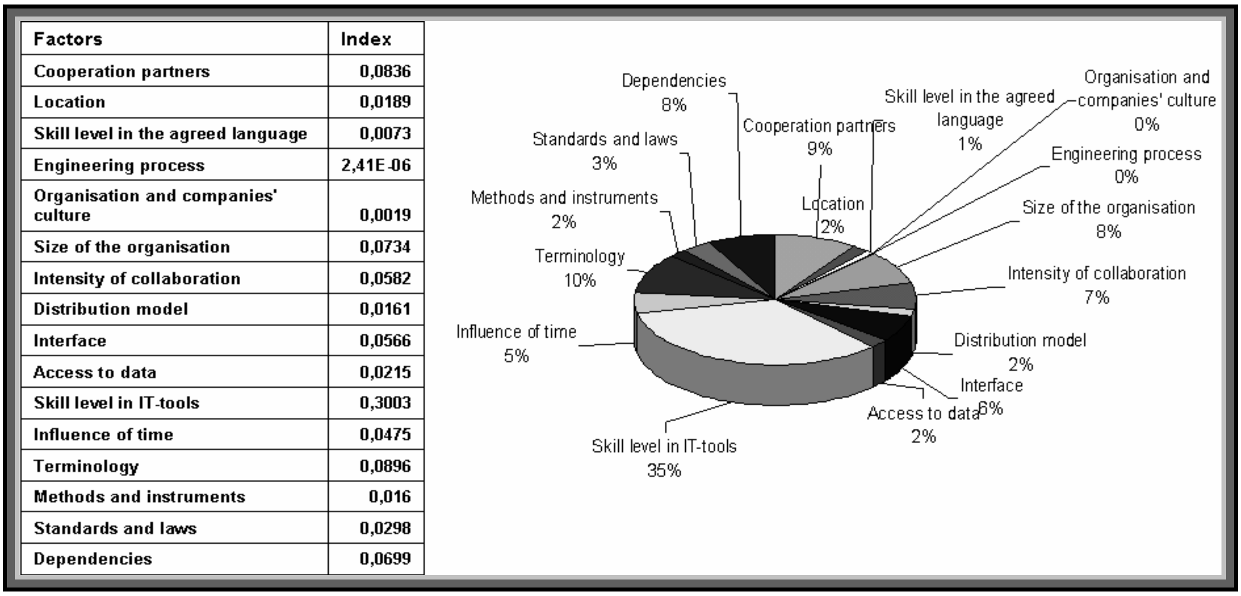
With respect to the hypothesis that all uncertain factors are susceptible to determination, the setting FP allows the identification of the factor that most deserves a better experimental measurement in order to reduce the target output uncertainty the most. Hence we can determine those factors that should be measured most precisely (cf. [Wagner, 2007]) in our empirical study.

There are various methods available for a sensitivity analysis. The Fourier Amplitude Sensitivity Test (FAST) is a commonly-used approach that is based on Fourier series describing the output functions (cf. [Wagner, 2007]). When results of the FAST are applied to our influencing factors, they not only give a qualitative ranking as, for example, the Morris method (cf. [Morris, 1991]), but even a quantification of the influencing factors.

We use the sensitivity analysis tool SimLab (cf. [SimLab, 2005]) for the analysis. For this we have to define all needed input parameters and their value sets. The tool then generates the samples needed for the FP analysis. Our goal is to determine with the help of the expert data which factors have the most impact on the choice of each tool.

The value sets of the influencing factors are derived from the results of the survey in [Laurent, 2007a]. For each tool and each value of a factor, the experts gave us a rating. These ratings compose the value sets of the factors. With the help of any simple model which uses each factor once, and with the same weighing, we can now determine the factors which have the most impact. Toward this end we have generated 50000 samples.

The FP setting gives us a ranking of the factors for each considered IT-tool. For example, Figure 4-23 shows the indices for the tool ‘Workflow System’. The factor ‘Skill level in IT-tools’ is responsible for 35% of the experts’ decision for this tool; the second most important factor is ‘Terminology’ with 10% and the third one ‘Cooperation partners’ with 9%.

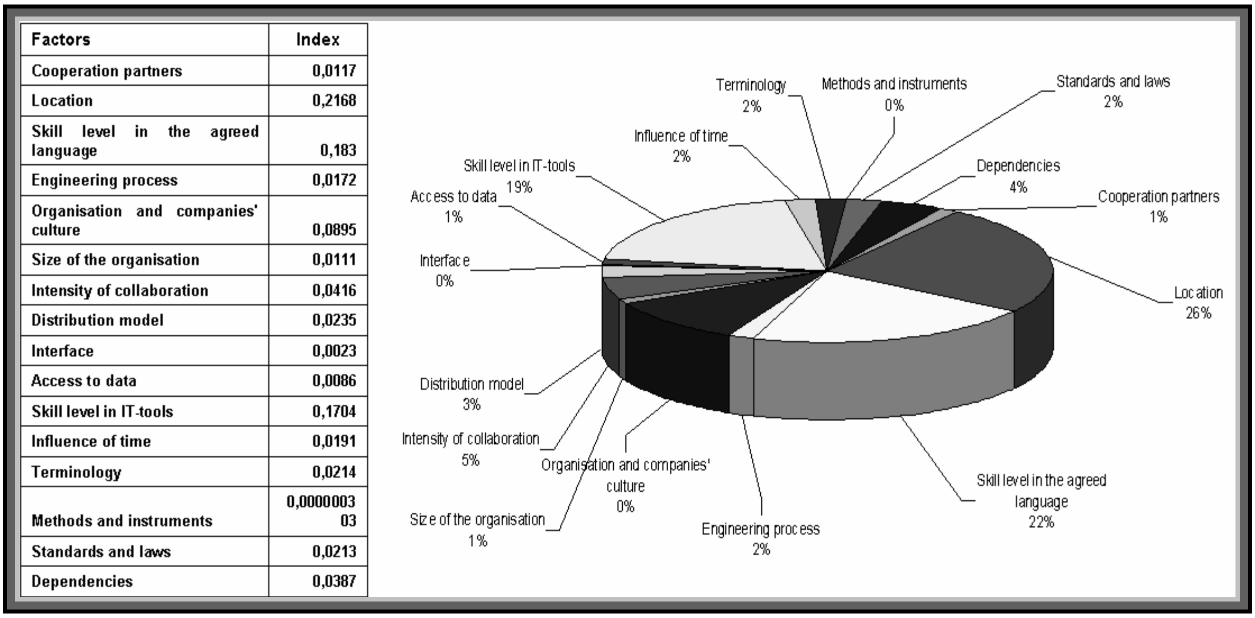


Factors	Index
Cooperation partners	0,0836
Location	0,0189
Skill level in the agreed language	0,0073
Engineering process	2,41E-06
Organisation and companies' culture	0,0019
Size of the organisation	0,0734
Intensity of collaboration	0,0582
Distribution model	0,0161
Interface	0,0566
Access to data	0,0215
Skill level in IT-tools	0,3003
Influence of time	0,0475
Terminology	0,0896
Methods and instruments	0,016
Standards and laws	0,0298
Dependencies	0,0699

Dependencies 8%
Skill level in the agreed language 1%
Organisation and companies' culture 0%
Standards and laws 3%
Cooperation partners 9%
Engineering process 0%
Methods and instruments 2%
Location 2%
Size of the organisation 8%
Terminology 10%
Intensity of collaboration 7%
Influence of time 5%
Distribution model 2%
Interface 6%
Access to data 2%
Skill level in IT-tools 35%

Figure 4-23: Factor prioritisation for the tool 'Workflow System'.

However, another example shows clearly that this factor prioritization is not the same for each tool. For example, for the tool 'Online-Conferencing': the factor 'Location' is responsible for 26% of the experts' decision for this tool; the second most important factor is 'Skill level in the agreed language' with 22% and the third one 'Skill level in IT-tools' with 19%.



Factors	Index
Cooperation partners	0,0117
Location	0,2168
Skill level in the agreed language	0,183
Engineering process	0,0172
Organisation and companies' culture	0,0895
Size of the organisation	0,0111
Intensity of collaboration	0,0416
Distribution model	0,0235
Interface	0,0023
Access to data	0,0086
Skill level in IT-tools	0,1704
Influence of time	0,0191
Terminology	0,0214
Methods and instruments	0,000000303
Standards and laws	0,0213
Dependencies	0,0387

Terminology 2%
Methods and instruments 0%
Standards and laws 2%
Influence of time 2%
Skill level in IT-tools 19%
Dependencies 4%
Access to data 1%
Cooperation partners 1%
Interface 0%
Location 26%
Distribution model 3%
Intensity of collaboration 5%
Organisation and companies' culture 0%
Skill level in the agreed language 22%
Size of the organisation 1%
Engineering process 2%

Figure 4-24: Factor prioritisation for the tool Online-Conferencing.

These two examples demonstrate that each tool has its own characteristics and that their implementation does not depend on the same boundary conditions of global development projects.

The entire results of the sensitivity analysis are shown in appendix 8.2 and have been presented in [Laurent, 2007b].

4.4.3. Decision Logic

The knowledge-base represents an evaluation of the experts for each tool and each characteristic of a factor, which corresponds to the usability of the tool. According to [ISO 9241, 1998] usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

Experts have given a usability assessment to each tool with respect to each value of the influencing factors. This usability assessment uses the same linguistic terms 'Not Adapted', 'Not recommended', 'Neutral', 'Recommended', and 'Very adapted' for each tool considered which are described in detail in the following section 4.5. Hence, for each tool we consider an LV O_{tool} (see also section 4.5).

As already mentioned in section 4.2.4.2 the inference process is carried out individually for each tool, so that our rule-base consider an IT-tool at the same time only. We have, for example, in our knowledge-base for the tool 'Instant messaging' the following rules:

(1) if 'Intensity of collaboration' is 'Low & irregular' then 'Instant messaging' is 'Not adapted'

(2) if 'Intensity of collaboration' is 'Integrated' then 'Instant messaging' is 'Recommended'

These two if-then clauses can also be formally rewritten as:

if $I_{f,i} = t_{f,i} \Rightarrow O_{tool,i} = t_{tool,i}$

Where:

- t_f is a term of the LV I_f and t_{tool} is a term of the LV O_{tool}.
- $i = \{1...r_{tool}\}$, with *i* being the index of rules for the considered tool. The number of rules for this tool is then r_{tool}.
- $f = \{1...f_{\max}\}$, with *f* being the index of influencing factors with $f_{\max}$ the number of influencing factors.

The expert-evaluation with respect to each single factor is not sufficient, however, to make a decision about the most appropriate collaboration-platform for a specific project. At this point, the inference process starts in order to take the entire project configuration into account. The decision logic is described in the following.

1st step: MIN (AND) inference

The usability of a tool does not depend on a single factor only, but rather on the global project configuration. Thus, we have several input variables $I_f : f = \{1 \ldots f_{\max}\}$. Consequently, in addition to the logical connective if-then, another logical operator is needed: AND. Then the if-then clauses are:

if $I_{1,i} = t_{1,i}$ and $I_{2,i} = t_{2,i}$ and … and $I_{f\max,i} = t_{f\max,i}$ then $O_{tool,i} = t_{tool,i}$

Where:

- t_f is a term of the LV I_f and t_{tool} is a term of the LV O_{tool}.
- $i = \{1 \ldots r_{tool}\}$, with *i* being the index of rules for the considered tool. The number of rules for this tool is then r_{tool}.
- $f = \{1 \ldots f_{\max}\}$, with *f* being the index of influencing factors with $f_{\max}$ the number of influencing factors.

This set of rules is the rule base of our tool. Thus, we follow now the methodology to express a rule base described in section 4.2.3.4, and start first with a fuzzy intersection operation or MIN inference (see section 4.2.3.3).

Note at this point that each influencing factor *f* has a single weight w.r.t. the tool, denoted by $w_{tool,f} \in R_+$, resulting from the sensitivity analysis described in section 4.4.2.

The MIN inference is then:

$$O_{tool,i} = \bigcap_{f=1}^{f_{\max}} w_{tool,f} I_{i,f}, \quad i = 1 \ldots r_{tool}$$

2nd step: MAX (OR) composition

To combine all rules we now use a fuzzy union or MAX composition as described in section 4.2.3.4. For one single tool, we have as many rules as possibilities to combine the terms of each factor.

The result of the MAX composition can be rewritten as:

$$O_{tool} = \bigcup_{i=1}^{r_{tool}} O_{tool,i}, \quad i = 1...r_{tool}$$

These steps are repeated for each tool under consideration.

4.5. Defuzzification

Defuzzification is carried out to convert the fuzzy output set to a „crisp“ number so that it can be easily readable and understandable. A „crisp“ value is synonymous with a “traditional” sharp value opposed to a fuzzy quantity.

The output set O$_{tool}$ represents the usability of the IT-tool under consideration regarding the actual project configuration. This set will be defuzzified to propose an appropriate collaboration-platform to the user. As already mentioned usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use (cf. [ISO 9241, 1998]).

The LV representing the output has been adapted from the one presented in [Laurent, 2006] and has the same definition for each single tool. The output for each tool is represented by a linguistic variable having the term set $T = \{$ $NotAdaptedAtAll, NotAdapted, Neutral, Adapted, VeryAdapted\}$ and the characteristics shown in Figure 4-25.

$Linguistic\ Variable : Output_{tool}$

$T = \{NotAdaptedAtAll, NotAdapted, Neutral, Adapted, VeryAdapted\}$

$Base: 0\ \text{-}\ 100\ (\%)$

$Assignment\ of\ the\ term\ set\ to\ the\ base:$

$Not\ adapted\ at\ all: \quad O_{tool,NAA} = \langle 0,16,0,16 \rangle_{LR}$

$Not\ adapted: \quad O_{tool,NA} = \langle 42,42,16,16 \rangle_{LR}$

$Neutral: \quad O_{tool,N} = \langle 58,58,16,16 \rangle_{LR}$

$Adapted: \quad O_{tool,A} = \langle 74,74,16,16 \rangle_{LR}$

$Very\ Adapted: \quad O_{tool,VA} = \langle 90,100,16,0 \rangle_{LR}$

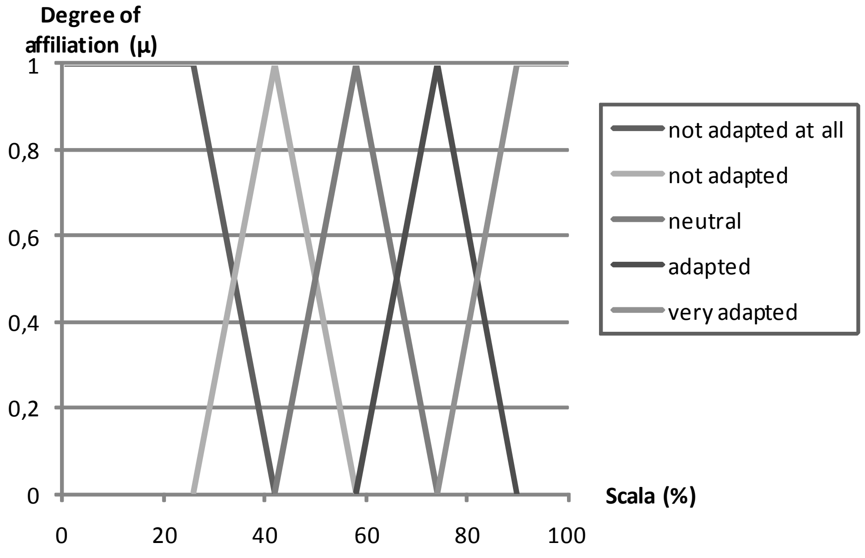


Figure 4-25: Representation of the linguistic variable ‘Output’ (in accordance with [Laurent, 2006]).

There is no general statement about ‘the right’ defuzzification method. Although defuzzification methods reflect different meanings, the fundamental objectives can approximately be differentiated to two goals: look for “the best compromise” or “the most plausible solution”.

The fuzzy output set gives for every possible value $x \in E$ (E = base of the output variable) a grade of membership, that describes to what extent this value *x* is reasonable to use. To use the fuzzy expert system as a decision-helper, we must transform this fuzzy information into a single value *x’* that will actually be applied. This transformation from a fuzzy set to a crisp number is called defuzzification.

The most common methods are maximum, moment, and center of area defuzzification. All defuzzification methods have their eligibility depending on the targeted application. The following presents the advantages and disadvantages of these methods.

Definition 17. Maximum defuzzification

Maxima methods are very quick and easy to use, since they consider only values with maximum membership. The most common one is the mean-of-maximum method. It takes the mean of those points where the membership function is at a maximum and thus can be written as:

$x' = \frac{\int_S x.dx}{\int_S dx}$, where $S = \{(x, y) | \mu(x) = y_0 \wedge y \leq y_0\}$ and $y_0 = \max_{x \in E} \mu(x)$ (see Figure 4-26)

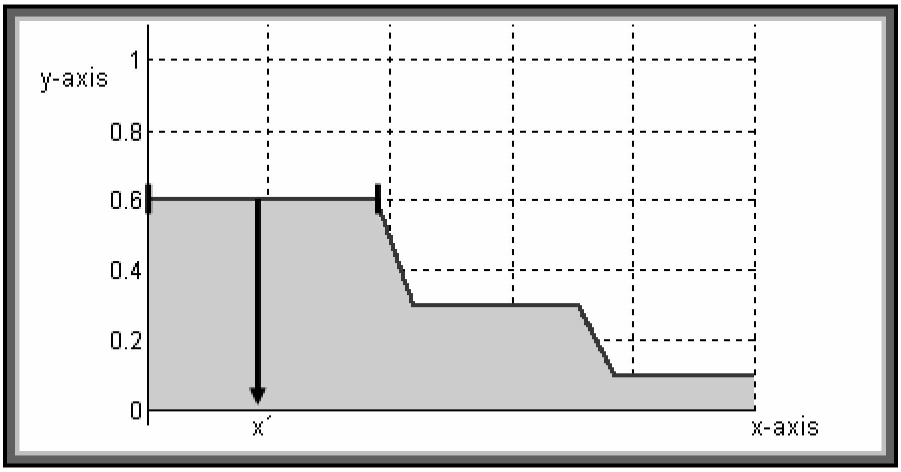


Figure 4-26: Maximum defuzzification.

This method does not work well in general. Especially in the following case (see Figure 4-27):

There exist *x* ranges where the *y* value is constant at the max value y_0 and other places where the maximum value is only reached for a single *x* value. When this happens the single value gets too much an influence in the defuzzified value.

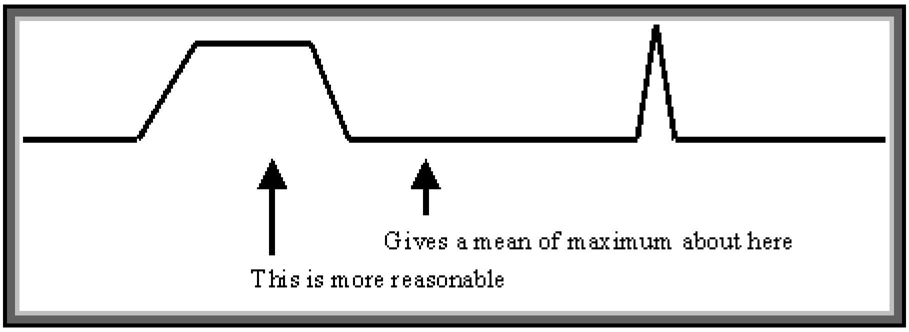


Figure 4-27: Problems of mean-of- maximum defuzzification (cf. [NRC, 2006]).

In fact, it is not concerned about a reasonable average value, since it only observes the maximum membership values and can be unstable.

Definition 18. Center of area defuzzification

This method calculates the centre of area under the membership function of the resulting fuzzy set.

$$x' = \frac{\int_U x.\mu(x).dx}{\int_U \mu(x).dx}$$, where *U* is the referential set of *x.* (see Figure 4-28)

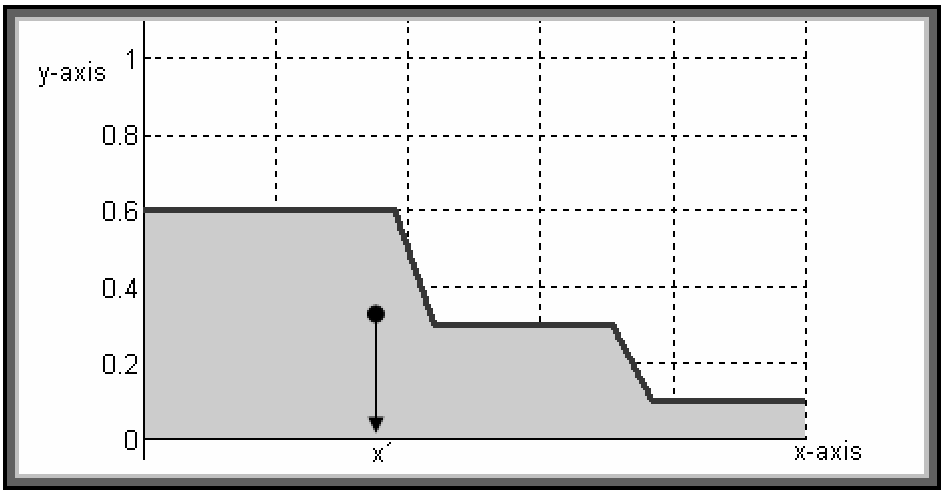


Figure 4-28: Center of area defuzzification.

It often gives a stable and well-balanced result, which is sufficient for most forms of fuzzy sets. However, a disadvantage of this technique is the high computational demand in computing for the areas under the membership functions.

Definition 19. Moment defuzzification

This method subdivides the fuzzy set into different shapes by partitioning it vertically at each salient point of the set (see Figure 4-29). The results are rectangles, triangles and trapezoids. The centre of gravity (moment) and area of each shape resulting from the partitioning is calculated using the appropriate formulas (cf. [NRC, 2006]). The first moment of area x' of the whole set corresponds to the defuzzification value and is then given by:

$x' = \frac{\sum_{i=1}^{n} x_i'.S_i}{\sum_{i=1}^{n} S_i}$, where x'_i is the local centre of gravity, S_i is the local area of each shape.

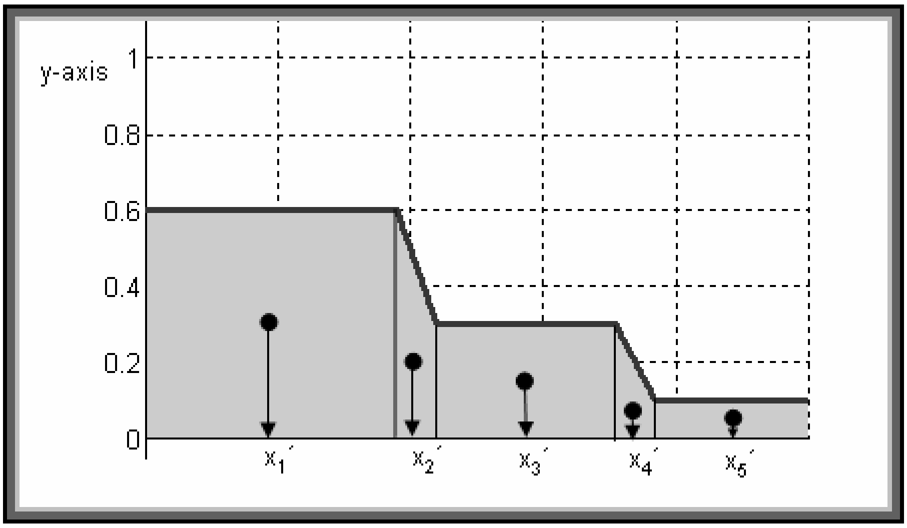


Figure 4-29: Moment defuzzification.

This method is similar to the center of area defuzzification and gives stable and well-balanced results, but is computationally faster and easier, since the areas under consideration are simplified.

Our fuzzy-expert-system does not only consider factors that make a good case for the implementation of a tool, but it considers the complete project configuration; this means also the critical factors. Therefore, we are not interested in the combination of tools and factors that reach a maximum only, but want to have a statement concerning the whole project configuration. Hence, the maximum defuzzification method is not appropriate for our needs.

Instead, we consider the following two methods of defuzzification:

- Center of area defuzzification
- Moment defuzzification

Therefore, our fuzzy-expert-system offers two proposals to configure an IT-collaboration-platform which present the results of both calculations: center of area and moment defuzzification. The user can choose the proposal most appropriate for his goal and thus takes the final decision. The two proposals are very similar, differing only slightly.

4.6. Implementation

The method to configure IT-collaboration-platforms has been used to develop a small application: the Collaboration-Platform Configurator (CPC). This tool is an easy-to-use version of the fuzzy-expert-system. This section describes the development of the CPC.

4.6.1.1. FuzzyJ Toolkit

The CPC has been programmed as a java application. The core processing is done by the FuzzyJ toolkit (cf. [NRC, 2006]). This Java application programming interface (API) enables the use of fuzzy logic and fuzzy reasoning with Java.

The toolkit consists of a set of classes that allows one to manipulate fuzzy information and construct fuzzy systems. The fundamental API elements are fuzzy variables, fuzzy sets and fuzzy values. A fuzzy variable is defined by its name, a base, and a set of fuzzy terms (see 4.2.3.1).

The toolkit uses the classes FuzzyVariable, FuzzySet, and FuzzyValue. A fuzzy term has no explicit representation, since each fuzzy term is defined by fuzzy values. The FuzzyJ Toolkit enables the construction of many fuzzy sets (see Figure 4-30). In our implementation we mainly use the simple singleton, triangle, rectangle, and left/right linear fuzzy sets.

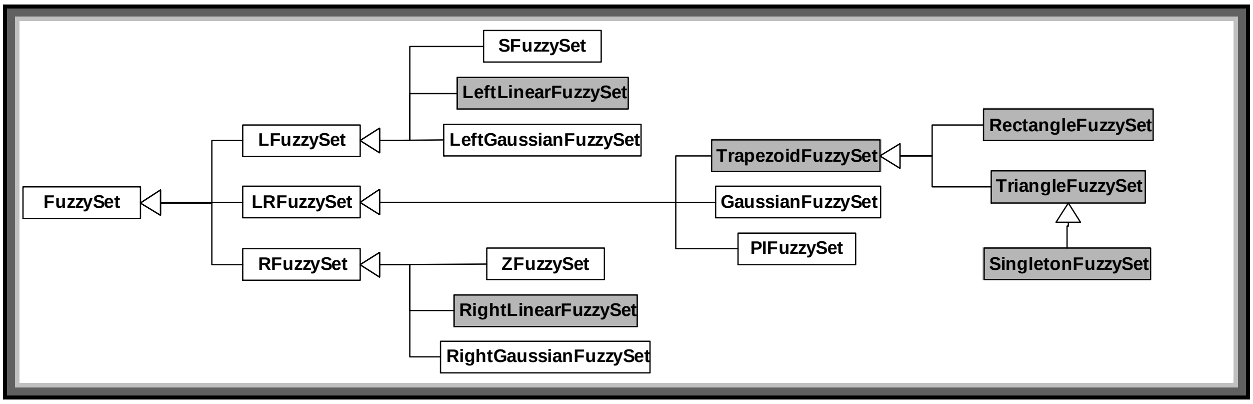


Figure 4-30: Inheritance structure of the class FuzzySet.

In addition, the FuzzyJ Toolkit has various processing functions representing the different defuzzification methods, among them the functions maximumDefuzzify, momentDefuzzify, centerOfAreaDefuzzify, and weightedAverageDefuzzify. These functions convert a FuzzyValue or a FuzzyValueVector into „crisp“ values.

4.6.1.2. Program Architecture

The fundamental program structure is modular. Consequently, data import, input and program control can be added or connected to the program core in arbitrary variants. For a correct handling only the interfaces and a program control need to be implemented. The package architecture is shown in Figure 4-31 and contains the following components:

- base: program control, system core, and prototype example
- base.data: resources, interface’s resources, and data elements like FuzzyElement for import
- base.input: input components, user input, and parser
- base.ui: images, scripts, styles, and ServerPages for prototype
- base.xml: interfaces for XML import

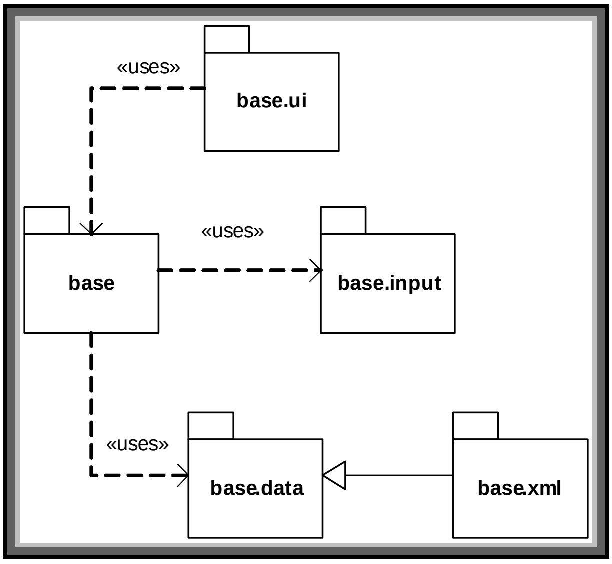


Figure 4-31: Package architecture.

4.6.1.3. Data

Nearly all data are available as an XML-file in the prototype; therefore, an XML document type definition (DTD) must be defined (see Figure 4-32). Information about the co-domain of the measured categories and their linguistic terms is defined in the DTD FuzzySetKnowledge. The expert data with weights of the factors resulting from the sensitivity analysis is defined in the DTD XPSKnowledge. The user input completes the CPC inputs and takes place via the graphic interface of the CPC.

The data classes are all derived from the abstract class FuzzyElement and use the class Resource. Resource is an interface of the data import. In the prototype we use XML data for data representation. Hence, the interface we have implemented and derived is called ResourceXml from the class Resource.

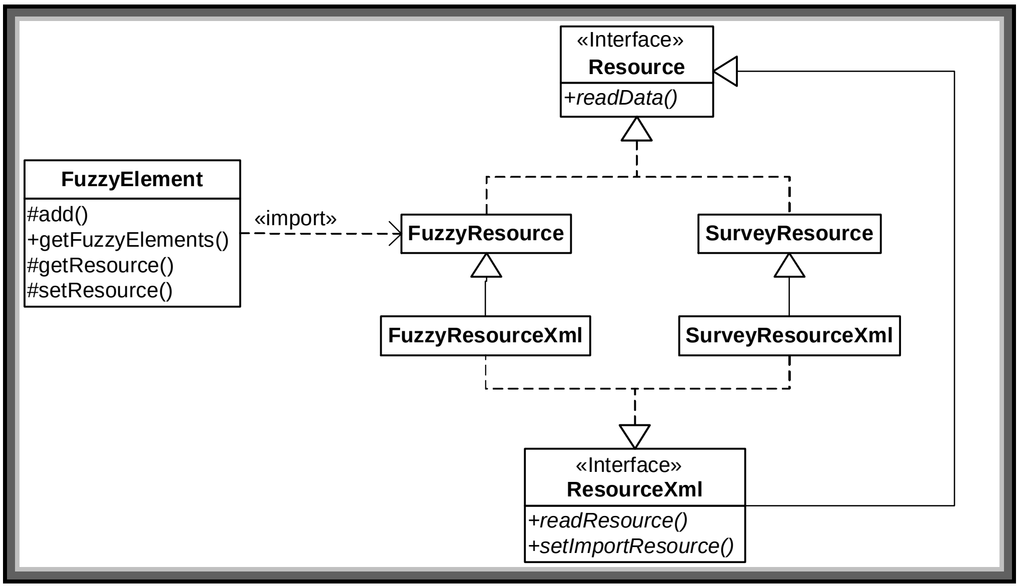


Figure 4-32: Data definition for XML data interface.

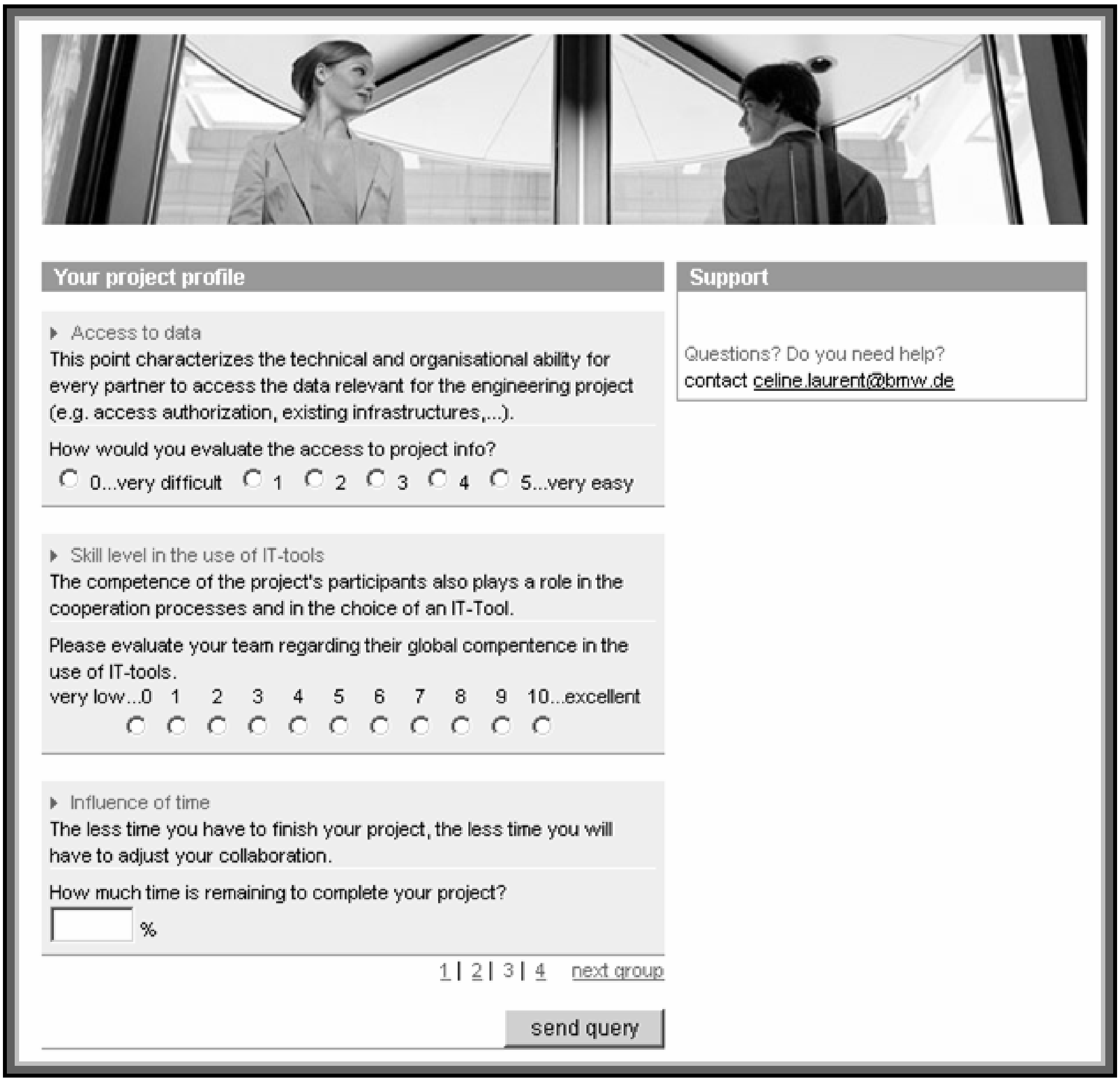
4.6.1.4. Program Logic

The program logic follows the methodology described in section 4.4.3. After the generation of the knowledge base using the results of the empirical study in [Laurent, 2007a] and the results of the sensitivity analysis (see sections 4.4.1 and 4.4.2), the reading of the fuzzy sets structure (see section 4.3), and the user inputs via the graphic interface, the procedure of evaluation starts following the steps described in 4.4.3.

4.6.1.5. User Interface

The user interface facilitates entering the current project configuration, as well as representing the results, so that everyone can understand and interpret the recommendation. It is of secondary importance whether the user interface takes on the form of a console application, an applet, or something else.

We have designed the prototype as a web application. The prototype uses Java Servlet technology, especially Java Server Pages (JSP). JSPs are a simple and fast solution to generate dynamic contents. Figure 4-33 shows a pair of the input masks presented to the user.



Your project profile

▸ Access to data

This point characterizes the technical and organisational ability for every partner to access the data relevant for the engineering project (e.g. access authorization, existing infrastructures,...).

How would you evaluate the access to project info?

○ 0...very difficult ○ 1 ○ 2 ○ 3 ○ 4 ○ 5...very easy

▸ Skill level in the use of IT-tools

The competence of the project's participants also plays a role in the cooperation processes and in the choice of an IT-Tool.

Please evaluate your team regarding their global compentence in the use of IT-tools.

very low...0	1	2	3	4	5	6	7	8	9	10...excellent
○	○	○	○	○	○	○	○	○	○	○

▸ Influence of time

The less time you have to finish your project, the less time you will have to adjust your collaboration.

How much time is remaining to complete your project?

□ %

1 | 2 | 3 | 4 next group

send query

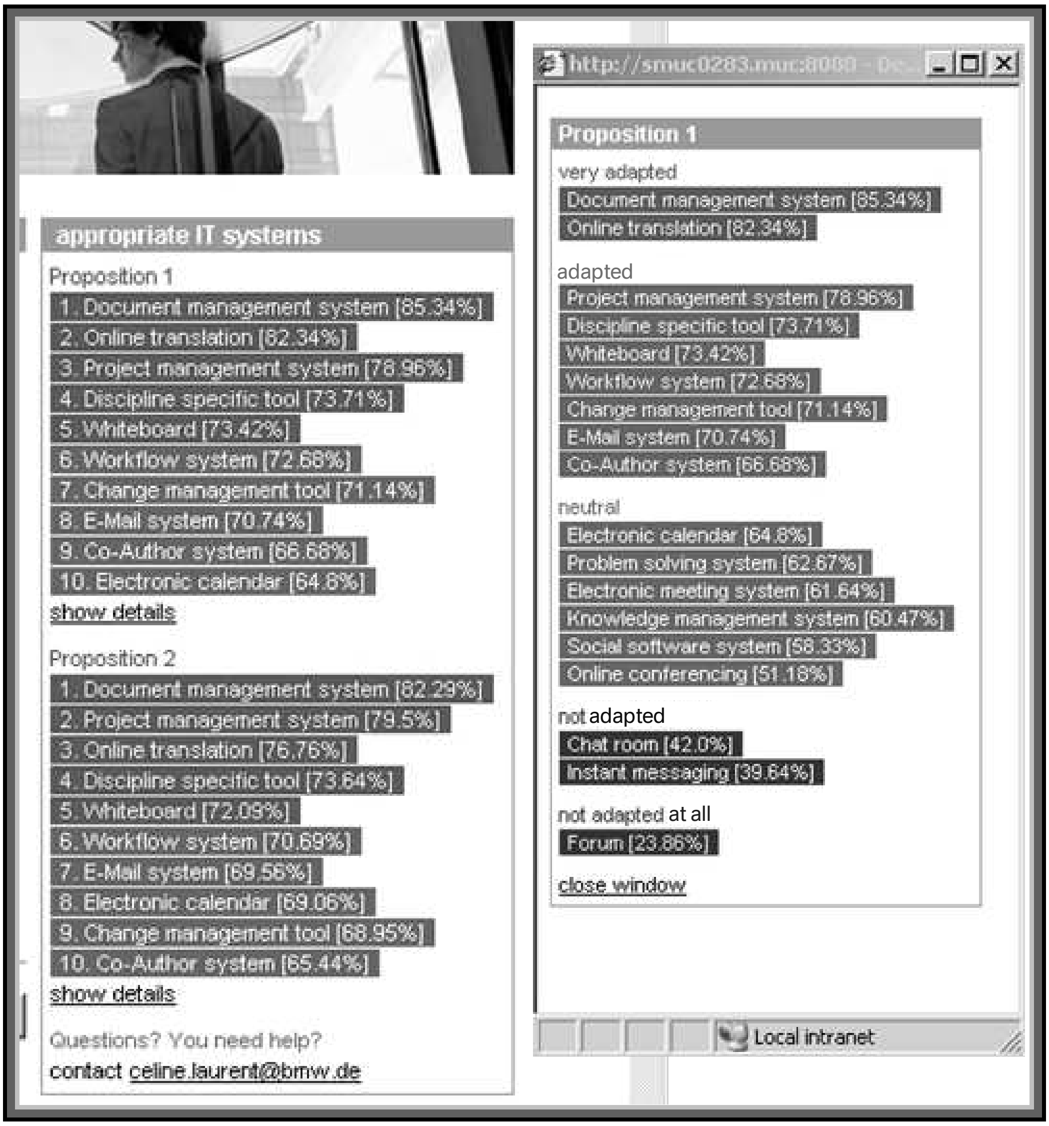
Support

Questions? Do you need help?

contact celine.laurent@bmw.de

Figure 4-33: Input mask of the CPC.

We implemented a java bean to make all functionalities available. Java beans are classes defining particular methods for event handling and getter/setter methods. The implemented bean adopts the program execution control. Error management and user interaction are basic tasks of the bean. Furthermore, it also carries out the session, program handling, and formatting of the return arguments of the system. If there is no error in the data import, initialisation and input of the user, the private function processSystem initiates the system calculations. The calculated results are returned in form of a map with all tools created with the function getResult(). The tools are sorted through the fields 'very adapted', 'adapted', 'neutral', 'not adapted' and 'not adapted at all' (see Figure 4-34).



appropriate IT systems

Proposition 1

1. Document management system [85.34%]
2. Online translation [82.34%]
3. Project management system [78.96%]
4. Discipline specific tool [73.71%]
5. Whiteboard [73.42%]
6. Workflow system [72.68%]
7. Change management tool [71.14%]
8. E-Mail system [70.74%]
9. Co-Author system [66.68%]
10. Electronic calendar [64.8%]

show details

Proposition 2

1. Document management system [82.29%]
2. Project management system [79.5%]
3. Online translation [76.76%]
4. Discipline specific tool [73.64%]
5. Whiteboard [72.09%]
6. Workflow system [70.69%]
7. E-Mail system [69.56%]
8. Electronic calendar [69.06%]
9. Change management tool [68.95%]
10. Co-Author system [65.44%]

show details

Questions? You need help?
contact celine.laurent@bmw.de

Proposition 1

very adapted

Document management system [85.34%]
Online translation [82.34%]

adapted

Project management system [78.96%]
Discipline specific tool [73.71%]
Whiteboard [73.42%]
Workflow system [72.68%]
Change management tool [71.14%]
E-Mail system [70.74%]
Co-Author system [66.68%]

neutral

Electronic calendar [64.8%]
Problem solving system [62.67%]
Electronic meeting system [61.64%]
Knowledge management system [60.47%]
Social software system [58.33%]
Online conferencing [51.18%]

not adapted

Chat room [42.0%]
Instant messaging [39.64%]

not adapted at all

Forum [23.86%]

close window

Figure 4-34: Output of the CPC.

4.7. Case Studies

In this section we will use the developed expert-system to configure IT-platforms for the support of existing development projects. With these case studies we principally want to verify the method, show its applicability to real projects, and highlight some advantages and disadvantages of this method.

A very important aspect when verifying a method is the necessity of having dissimilar projects. That is the case here. On the one hand, the method will be applied to the development of the tuner, a part of a car entertainment system, and an engine development project, and on the other hand, to a software development project.

4.7.1. Engineering Projects in the Automotive Field

The two engineering projects at BMW haven't been chosen randomly to verify the method of configuring IT-collaboration-platforms. In fact, both project managers have requested our help to configure an appropriate platform. First, the projects will be described with respect to the factors presented in 2.4; second, propositions of possible configurations for IT-systems will be determined with the help of the expert-system presented in 4, and third, the generated solution will be implemented.

4.7.1.1. Tuner-Development

Description of the project

A tuner is an electronic receiver that detects, demodulates and amplifies transmitted signals, for example, for an audio system. Due to its complex system integration, the development of the tuner transpires in an entangled cross-domain and -enterprise context.

In fact, the tuner is not an isolated component, but rather depends on many other components of BMW entertainment systems. To understand which role the tuner plays in the entire system, we take a look at the input/output diagram of the SDARS Tuner (Satellite Digital Audio Radio System) (see Figure 4-35).

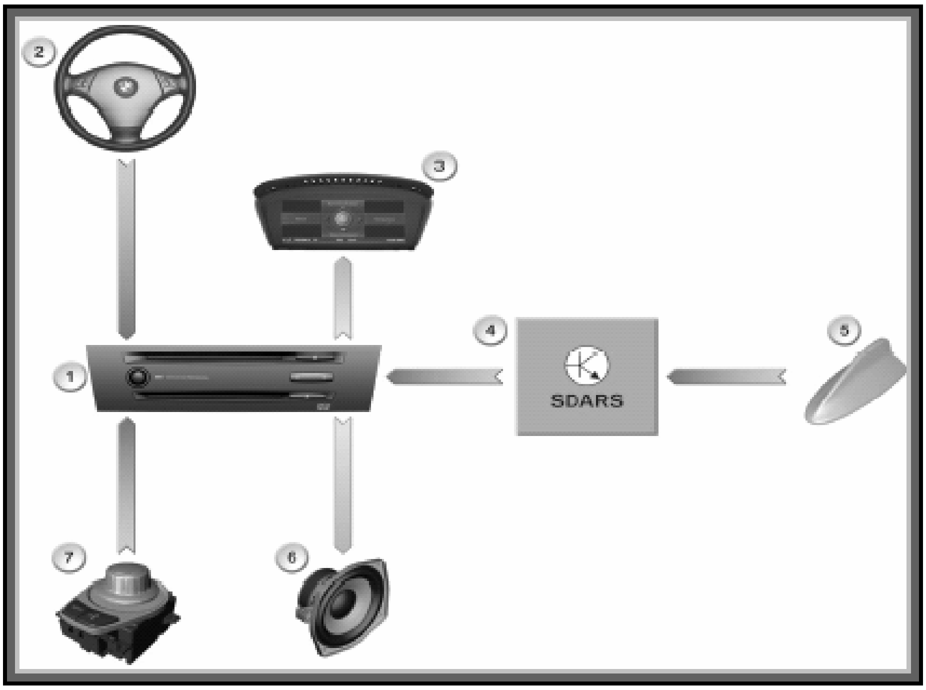


Figure 4-35: Input/Output diagram of the tuner SDARS.

The driver can work the Head Unit (HU, 1) via commands at the wheel (2) or via the controller (7). The roof aerial (5) receives the digital radio signals and transmits it to the tuner (4) that demodulates and amplifies the signals. The navigation system permits then the driver to listen to his favourite programs (audio speaker, 6) and shows it on the Central Information Display (CID, 3). All these components communicate with each other via CAN-Bus (Controller Area Network) or MOST-Bus (Media Oriented System Transportation).

Due to the different transmission protocols emitted in each country, it is necessary to test the tuner in all parts of the world. Presently, for example, the SDARS-tuner can only receive the emitted signals in the USA. In the future it must be reckoned with an increase of such country-specific transmission protocols. As a consequence it is inevitable that this development results in the collaboration of geographically distributed teams. Hence it's interesting to study this engineering process, since it's a complex interdisciplinary process with principal domains, such as communications-, software engineering, electronic and project management. Furthermore, since it's a distributed process, a collaboration platform makes sense.

To include the distribution aspect in the analysis of this interdisciplinary process, the V-Model has been instantiated in order to identify the collaboration needs of the engineering process (see Figure 4-36).

The development of the tuner is carried out by a specific supplier. BMW is in charge of the project management at the system level. BMW defines the requirements and elaborate a concept with the supplier. At the software level the requirements are also defined in collaboration with the supplier. At this point the supplier is able to design the software and implement it on his own. Then we enter the testing phase. Firstly, the SW-modules are tested by the supplier, secondly, an integration test takes place at BMW.

It's rather easy to identify when collaboration is needed between BMW and its supplier. In fact, we can identify, on the one hand, a collaboration need when defining requirements at system and software level and, on the second hand, to discuss the tests (integration tests, system tests and acceptance tests) and, eventually, to change the requirements and start the development of a new version. Figure 4-36 displays the collaboration needs in the various phases of the process.

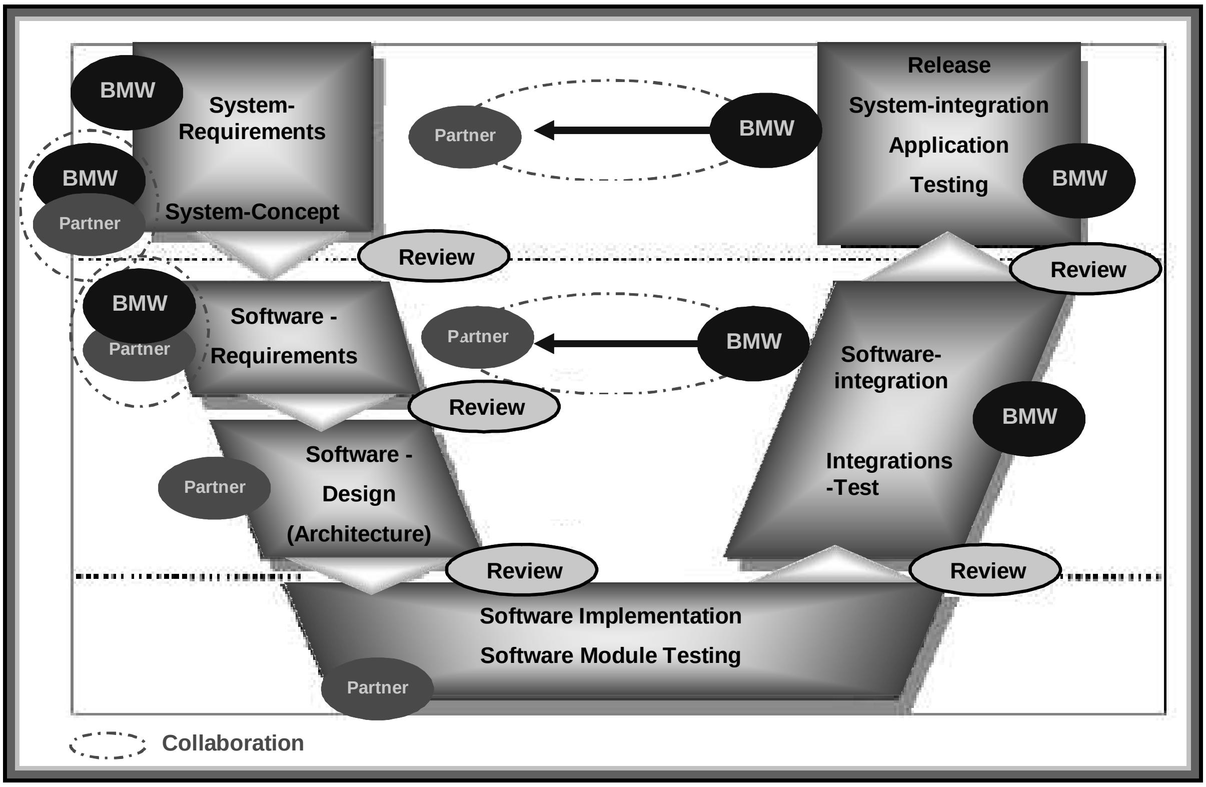


Figure 4-36: V-Model with identification of collaboration needs.

To configure an optimal collaboration-platform we will now use the developed method.

Application of the method

First, we have to describe the project with respect to the influencing factors given in sections 2.4 and 4.3. Toward this end we must remember that with fuzzy logic the assignment of any value or evaluation level of membership is open to subjective judgment. Fuzziness in assignment is okay, and it is natural (cf. [Treadwell, 1995]). Thus, we will use the set of natural-language words mentioned in section 4.2.3.1.

The number of cooperation partners is greater than five. In fact, the supplier developing the tuner, BMW Germany and North America as well as the suppliers of the other components of the entertainment system have to collaborate via this platform. The participants are all from different companies all of which are in the category of large-scale enterprises and located either in Germany or in the USA (by BMW). For this reason they speak German and/or English rather well.

Generally speaking, the different partners work sequentially, i.e. they build upon each other's work. Nevertheless, the collaboration is quite integrated in the daily work, especially because of the strong dependence of the tuner on the other components.

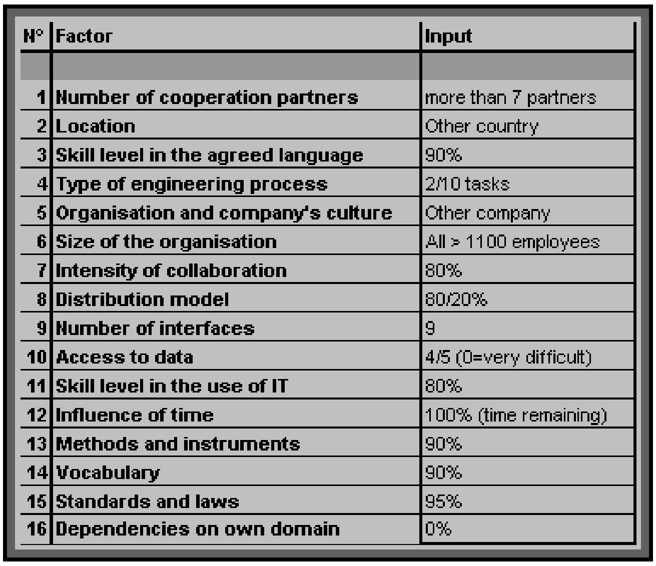
The tasks are somewhat equally distributed between the different suppliers; however, BMW only carries out the project and requirement management.

While there are very few organisational interfaces, technical interfaces are quite striking, especially in the case of overseas tests. The technical and organisational ability for every partner to access the relevant data is very good. A database containing all information is accessible to every participant.

The skill level of the participants in the use of IT is quite good. In fact, although it is not their speciality, in the majority of cases they have IT background and can easily deal with it. Since the development of the tuner is a continuous and enduring project, it is not really time-critical.

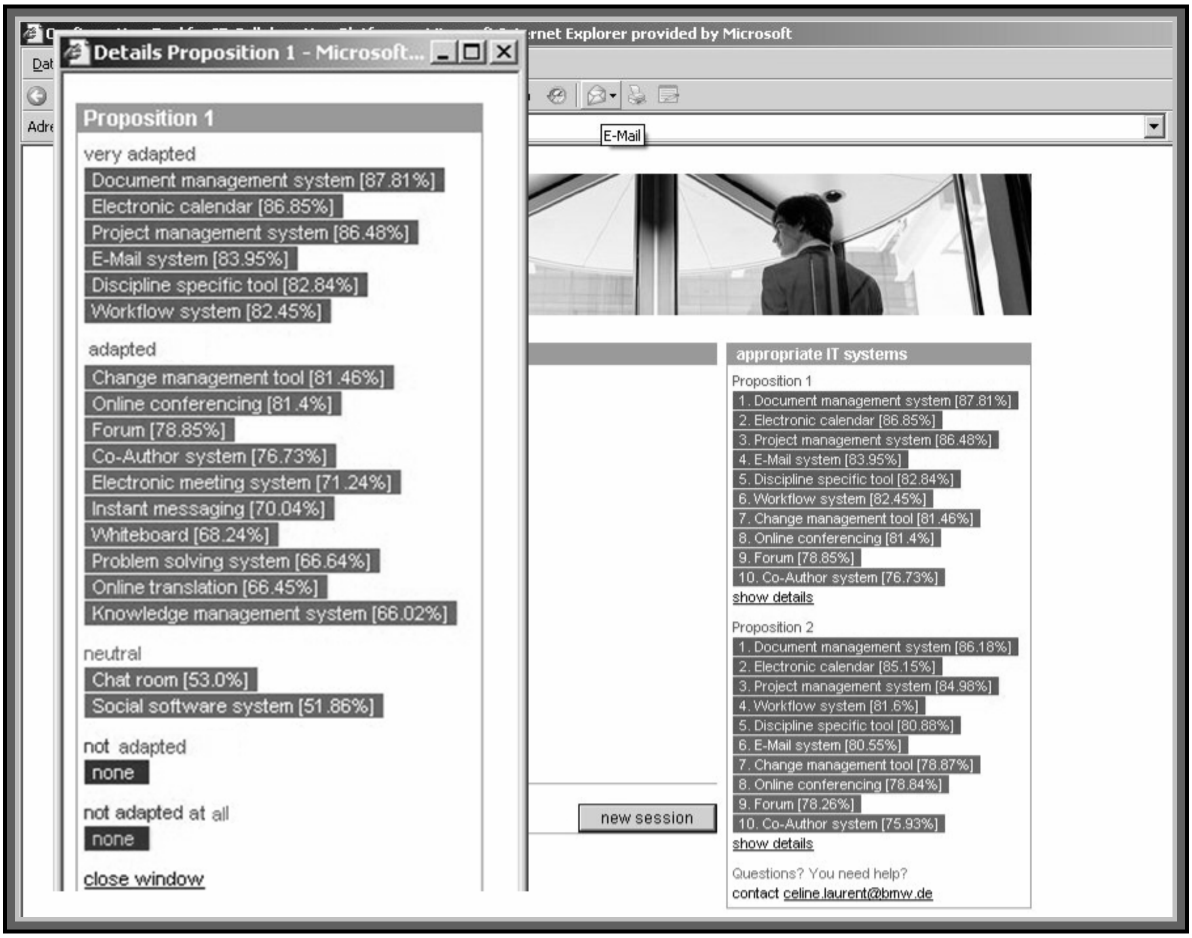
The vocabularies employed by the participants are generally the same. They are all specialists in the same fields of works. They also use the same methods and instruments and follow the same standards imposed by BMW. The participants can act quite independently.

We now use the method described in Chapter 4 to propose an appropriate configuration of IT-platform. We use the CPC described in 4.6. Figure 4-37 shows the input of this project, and respectively, Figure 4-38 the propositions made by the CPC.



N°	Factor	Input
1	Number of cooperation partners	more than 7 partners
2	Location	Other country
3	Skill level in the agreed language	90%
4	Type of engineering process	2/10 tasks
5	Organisation and company's culture	Other company
6	Size of the organisation	All > 1100 employees
7	Intensity of collaboration	80%
8	Distribution model	80/20%
9	Number of interfaces	9
10	Access to data	4/5 (0=very difficult)
11	Skill level in the use of IT	80%
12	Influence of time	100% (time remaining)
13	Methods and instruments	90%
14	Vocabulary	90%
15	Standards and laws	95%
16	Dependencies on own domain	0%

Figure 4-37: Input corresponding to the tuner-development.


rnet Explorer provided by Microsoft
Details Proposition 1 - Microsoft...
E-Mail
new session

Proposition 1

very adapted

- Document management system [87.81%]
- Electronic calendar [86.85%]
- Project management system [86.48%]
- E-Mail system [83.95%]
- Discipline specific tool [82.84%]
- Workflow system [82.45%]

adapted

- Change management tool [81.46%]
- Online conferencing [81.4%]
- Forum [78.85%]
- Co-Author system [76.73%]
- Electronic meeting system [71.24%]
- Instant messaging [70.04%]
- Whiteboard [68.24%]
- Problem solving system [66.64%]
- Online translation [66.45%]
- Knowledge management system [66.02%]

neutral

- Chat room [53.0%]
- Social software system [51.86%]

not adapted

- none

not adapted at all

- none

close window

appropriate IT systems

Proposition 1

1. Document management system [87.81%]
2. Electronic calendar [86.85%]
3. Project management system [86.48%]
4. E-Mail system [83.95%]
5. Discipline specific tool [82.84%]
6. Workflow system [82.45%]
7. Change management tool [81.46%]
8. Online conferencing [81.4%]
9. Forum [78.85%]
10. Co-Author system [76.73%]

show details

Proposition 2

1. Document management system [86.18%]
2. Electronic calendar [85.15%]
3. Project management system [84.98%]
4. Workflow system [81.6%]
5. Discipline specific tool [80.88%]
6. E-Mail system [80.55%]
7. Change management tool [78.87%]
8. Online conferencing [78.84%]
9. Forum [78.26%]
10. Co-Author system [75.93%]

show details

Questions? You need help?
contact celine.laurent@bmw.de

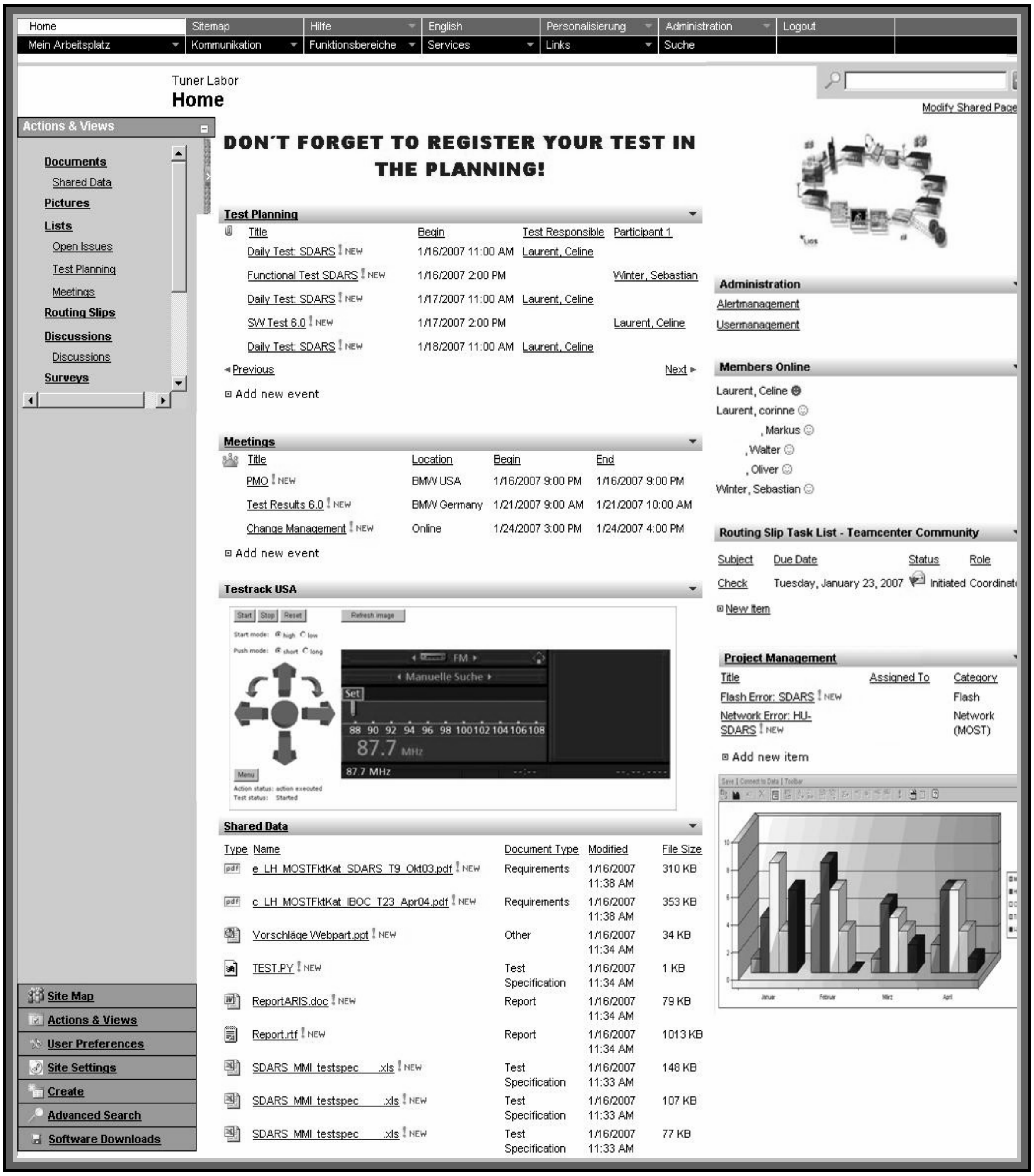
Figure 4-38: Proposed configuration for the tuner-development project.

Realisation

The proposition has been used to build an IT-platform based on the Virtual Project Space (VPS). VPS is a web-application for collaborative engineering integrated in the Partner Portal of the BMW Group (collaboration portal for BMW partners), which provides a platform with scalable collaboration functionality that facilitates real-time synchronous and asynchronous engineering collaboration between internal teams, as well with external suppliers and partners. It mainly supports management, informative and communication processes with functions like workflows, tasks-management, conferencing, shared folders and so on.

VPS is based on the software Teamcenter Community from UGS. Teamcenter Community is using the Microsoft SharePoint Technology. It provides features through the utilisation of modules, called Web Parts, which offer the possibility of showing information in various ways and organising it differently. The web parts can be configured according to the project's needs. It is also possible to develop new web parts that can then be easily integrated into the platform.

We have configured the VPS according to Figure 4-38 with a document management system, an electronic calendar, a project management system, a discipline specific tool and a workflow system. E-Mails are still appropriate to support the communication on this project and could also be integrated into the platform through ‘My Inbox Web Part’, allowing every participant to have its Project-E-Mails centralized on the platform (cf. [Microsoft, 2004]). Figure 4-39 presents the resulting collaboration-platform.



Tuner Labor

Home

DON'T FORGET TO REGISTER YOUR TEST IN THE PLANNING!

Test Planning

Title	Begin	Test Responsible	Participant 1
Daily Test: SDARS ! NEW	1/16/2007 11:00 AM	Laurent, Celine	
Functional Test SDARS ! NEW	1/16/2007 2:00 PM		Winter, Sebastian
Daily Test: SDARS ! NEW	1/17/2007 11:00 AM	Laurent, Celine	
SW Test 6.0 ! NEW	1/17/2007 2:00 PM		Laurent, Celine
Daily Test: SDARS ! NEW	1/18/2007 11:00 AM	Laurent, Celine	

◂ Previous Next ▸

⊡ Add new event

Meetings

Title	Location	Begin	End
PMO ! NEW	BMW USA	1/16/2007 9:00 PM	1/16/2007 9:00 PM
Test Results 6.0 ! NEW	BMW Germany	1/21/2007 9:00 AM	1/21/2007 10:00 AM
Change Management ! NEW	Online	1/24/2007 3:00 PM	1/24/2007 4:00 PM

⊡ Add new event

Testrack USA

Action status: action executed
Test status: Started

Shared Data

Type	Name	Document Type	Modified	File Size
pdf	e_LH_MOSTFktKat_SDARS_T9_Okt03.pdf ! NEW	Requirements	1/16/2007 11:38 AM	310 KB
pdf	c_LH_MOSTFktKat_IBOC_T23_Apr04.pdf ! NEW	Requirements	1/16/2007 11:38 AM	353 KB
	Vorschläge Webpart.ppt ! NEW	Other	1/16/2007 11:34 AM	34 KB
	TEST.PY ! NEW	Test Specification	1/16/2007 11:34 AM	1 KB
	ReportARIS.doc ! NEW	Report	1/16/2007 11:34 AM	79 KB
	Report.rtf ! NEW	Report	1/16/2007 11:34 AM	1013 KB
	SDARS_MMI_testspec_____.xls ! NEW	Test Specification	1/16/2007 11:33 AM	148 KB
	SDARS_MMI_testspec_____.xls ! NEW	Test Specification	1/16/2007 11:33 AM	107 KB
	SDARS_MMI_testspec_____.xls ! NEW	Test Specification	1/16/2007 11:33 AM	77 KB

Administration

Alertmanagement

Usermanagement

Members Online

Laurent, Celine ●
Laurent, corinne ☺
, Markus ☺
, Walter ☺
, Oliver ☺
Winter, Sebastian ☺

Routing Slip Task List - Teamcenter Community

Subject	Due Date	Status	Role
Check	Tuesday, January 23, 2007	Initiated	Coordinat

⊡ New Item

Project Management

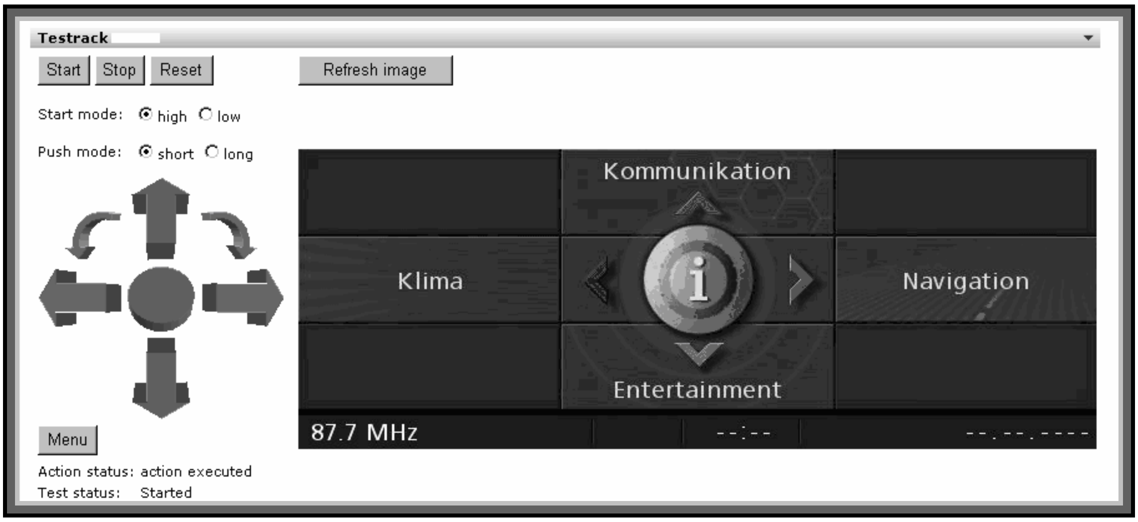
Title	Assigned To	Category
Flash Error: SDARS ! NEW		Flash
Network Error: HU-SDARS ! NEW		Network (MOST)

⊡ Add new item

Figure 4-39: Proposed collaboration-platform for the tuner-development project.

As mentioned above, at the moment the SDARS-tuner can only receive the emitted signals in the USA. The discipline-specific tool aims at testing the tuner via the platform without travelling, thus reducing travel costs and the time involved in development.

Subsequently, we programmed a new Web Part (see Figure 4-40) which enables the online and distributed testing of the tuner via internet.



Testrack

Start mode: ● high ○ low

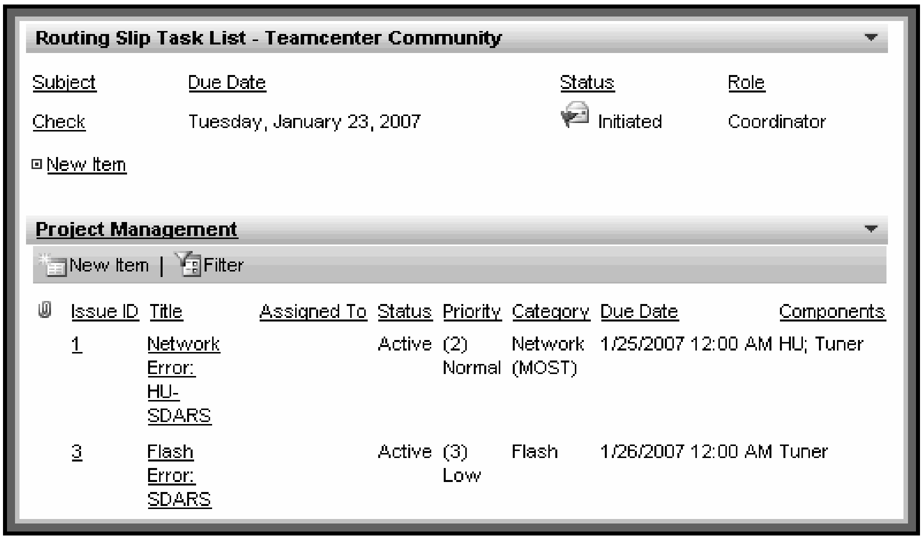
Push mode: ● short ○ long

Action status: action executed

Test status: Started

Figure 4-40: Web Part to execute distributed tests of the tuner via VPS.

Figure 4-41 shows the project management system with workflow implemented in VPS. With the help of this tool they can follow everyone's tasks and responsibility in the project. It also enables the categorisation of tasks and problems, the creation of statistics (see Figure 4-39) and the generation of workflows.



Routing Slip Task List - Teamcenter Community

Subject	Due Date	Status	Role
Check	Tuesday, January 23, 2007	Initiated	Coordinator

⊞ New Item

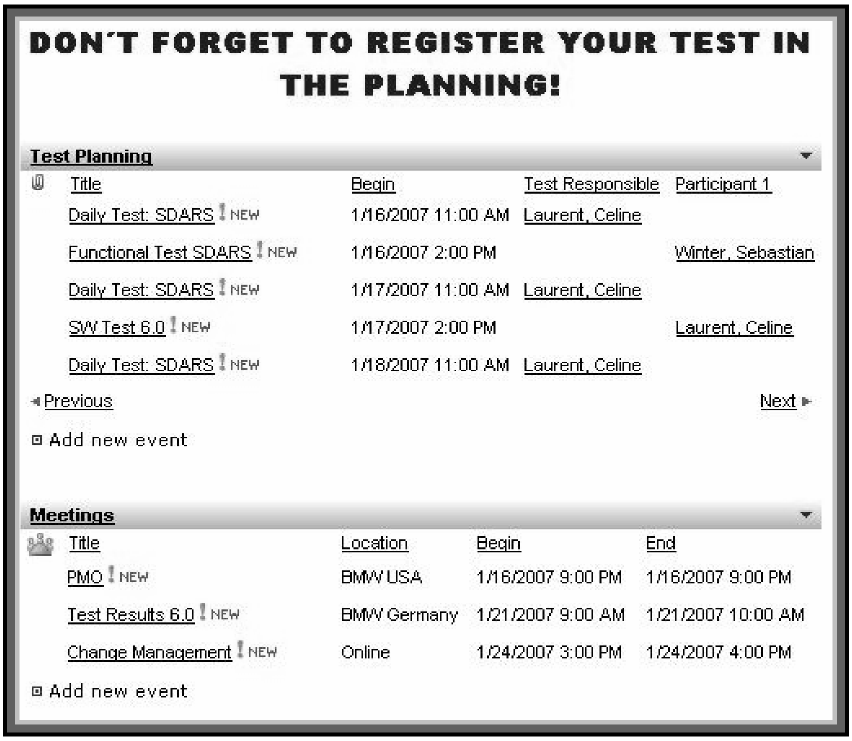
Project Management

New Item | Filter

	Issue ID	Title	Assigned To	Status	Priority	Category	Due Date	Components
	1	Network Error: HU-SDARS		Active	(2) Normal	Network (MOST)	1/25/2007 12:00 AM	HU; Tuner
	3	Flash Error: SDARS		Active	(3) Low	Flash	1/26/2007 12:00 AM	Tuner

Figure 4-41: Web Part for project management with workflow.

Figure 4-42 shows the electronic calendar implemented in the platform for the planning of the tuner tests as well as the project meetings.



DON'T FORGET TO REGISTER YOUR TEST IN THE PLANNING!

Test Planning

Title	Begin	Test Responsible	Participant 1
Daily Test: SDARS ! NEW	1/16/2007 11:00 AM	Laurent, Celine	
Functional Test SDARS ! NEW	1/16/2007 2:00 PM		Winter, Sebastian
Daily Test: SDARS ! NEW	1/17/2007 11:00 AM	Laurent, Celine	
SW Test 6.0 ! NEW	1/17/2007 2:00 PM		Laurent, Celine
Daily Test: SDARS ! NEW	1/18/2007 11:00 AM	Laurent, Celine	

◂ Previous　　Next ▸

⊡ Add new event

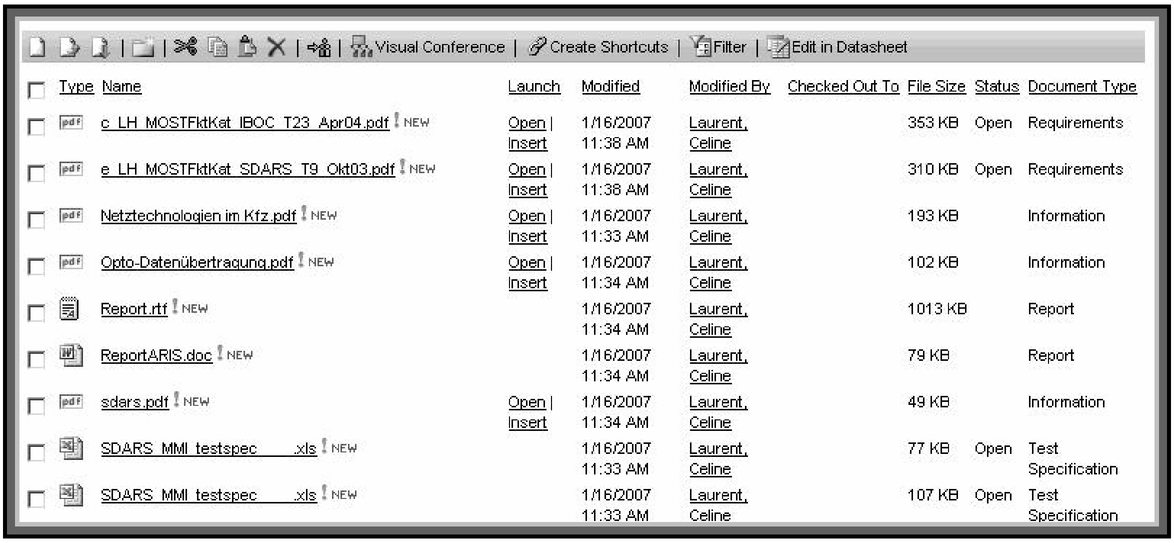
Meetings

Title	Location	Begin	End
PMO ! NEW	BMW USA	1/16/2007 9:00 PM	1/16/2007 9:00 PM
Test Results 6.0 ! NEW	BMW Germany	1/21/2007 9:00 AM	1/21/2007 10:00 AM
Change Management ! NEW	Online	1/24/2007 3:00 PM	1/24/2007 4:00 PM

⊡ Add new event

Figure 4-42: Calendaring Web Parts.

Figure 4-43 represents the document management system integrated in the platform which enables the creation of metadata, version control of documents, check-in/out functionalities, security and alert management.



Visual Conference | Create Shortcuts | Filter | Edit in Datasheet

□	Type	Name	Launch	Modified	Modified By	Checked Out To	File Size	Status	Document Type
□	pdf	c LH MOSTFktKat IBOC T23 Apr04.pdf ! NEW	Open | Insert	1/16/2007 11:38 AM	Laurent, Celine		353 KB	Open	Requirements
□	pdf	e LH MOSTFktKat SDARS T9 Okt03.pdf ! NEW	Open | Insert	1/16/2007 11:38 AM	Laurent, Celine		310 KB	Open	Requirements
□	pdf	Netztechnologien im Kfz.pdf ! NEW	Open | Insert	1/16/2007 11:33 AM	Laurent, Celine		193 KB		Information
□	pdf	Opto-Datenübertragung.pdf ! NEW	Open | Insert	1/16/2007 11:34 AM	Laurent, Celine		102 KB		Information
□		Report.rtf ! NEW		1/16/2007 11:34 AM	Laurent, Celine		1013 KB		Report
□		ReportARIS.doc ! NEW		1/16/2007 11:34 AM	Laurent, Celine		79 KB		Report
□	pdf	sdars.pdf ! NEW	Open | Insert	1/16/2007 11:34 AM	Laurent, Celine		49 KB		Information
□		SDARS MMI testspec .xls ! NEW		1/16/2007 11:33 AM	Laurent, Celine		77 KB	Open	Test Specification
□		SDARS MMI testspec .xls ! NEW		1/16/2007 11:33 AM	Laurent, Celine		107 KB	Open	Test Specification

Figure 4-43: Web Part for document management.

4.7.1.2. Engine-Development

Description of the project

The engine-development project is the result of a cooperation between two car producers, BMW and another European OEM. Together, they are building a four-cylinder spark-ignition engine that will be used for small and medium class vehicles.

At the beginning of the project, the different tasks were distributed between the two partners depending on the strength of each for each specific issue. BMW had the Know-How for the innovation, especially for the VALVETRONIC-technology based on the principle of a continuously variable adjustment of the cylinder filling dependent on the valve timing. Hence, BMW is responsible for the design, development and conception. The other OEM has a great deal of experience in the mass production of 4-cylinder engines and is subsequently responsible for purchasing and production. Thereafter, the integration in the corresponding vehicle is done on their respective assembly line, in England for BMW and in France for the second OEM.

The official project space is located in Munich and consists of participants of BMW and the other OEM. The employees of the partner OEM are in Munich to assure, on the one hand, the interface with the employees of their original company and, on the other hand, to acquire a technical comprehension of the developed technology. A real collaboration between the two partners is very difficult to obtain, in fact their guidelines, standards, methods, and instruments are very different. Consequently, they chose a SE-Team structure (Simultaneous Engineering).

The term Simultaneous Engineering (SE) means that the different sub-processes are worked out parallelly. Instead of using final results to begin a new process, intermediate results are used. This type of engineering process is time-saving compared to a classical development process. However, this process implies other difficulties like following the global development due to complex dependencies between the sub-processes. Thus, an intensive collaboration and coordination between each team is required.

The structure of an SE-Team is not hierarchic, but rather is articulated in a network. The SE-Team consists of a team leader and at least one specialist from each domain in order to make decisions with respect to the requirements of each domain. The domains represented depend on the actual phase of the project and on which component the Team is working, but are typically: calculation, construction, function, material, test, procurement, purchasing, quality and production.

Each SE-Team works on a subsystem of the engine and works together with the other SE-Teams in order to achieve the best results. In this kind of organisation the different project partners must be open-minded to suggestions and critiques from other SE-Teams.

At the end, the work of all SE-Teams together constitutes the engine which will be integrated into the MINI for BMW. Some function and engine tests have already been done during the entire development to make sure that the assembly will work, a procedure which demands regular feedbacks and collaboration.

Application of the method

First, we describe the project with respect to the influencing factors described in sections 2.4 and 4.3. Toward this end we will also use the set of natural-language words mentioned in section 4.2.3.1.

The main cooperation partners are BMW and another European OEM. Both of them have, of course, suppliers involved in the development process, but they are not involved at the global cooperation level. The suppliers only have bi-directional interfaces to their respective OEM.

Thus, the two partners are from different companies with a very marked culture, which are in the category of large-scale enterprises and are located either in Germany, in England (by BMW) or in France. Their level in the agreed project language is more or less good.

The participants work on their tasks concurrently. This is in fact one of the main characteristics of SE-Teams. For the same reason the collaboration is quite integrated into the daily work. The tasks are completely equally distributed between the two OEMs.

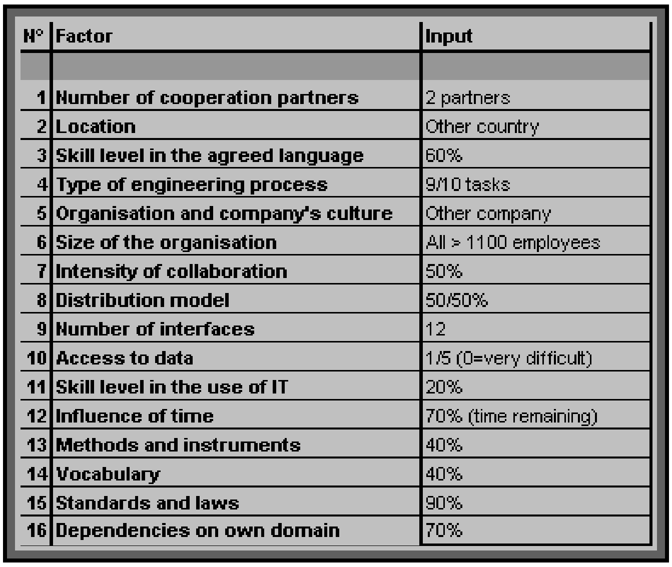
Organisational interfaces as well as technical interfaces are quite striking. In fact, the two partners are competing enterprises, which complicates their relationship, flexibility and facilities. The technical and organisational ability to access the relevant data is therefore also very restricted.

The skill level of the participants in the use of IT is quite bad. In fact, it is not their speciality, and very few participants can really deal with IT.

The development of this engine is a single project and is considered to be time-critical, since it could have a lot of repercussions on the companies' profits.

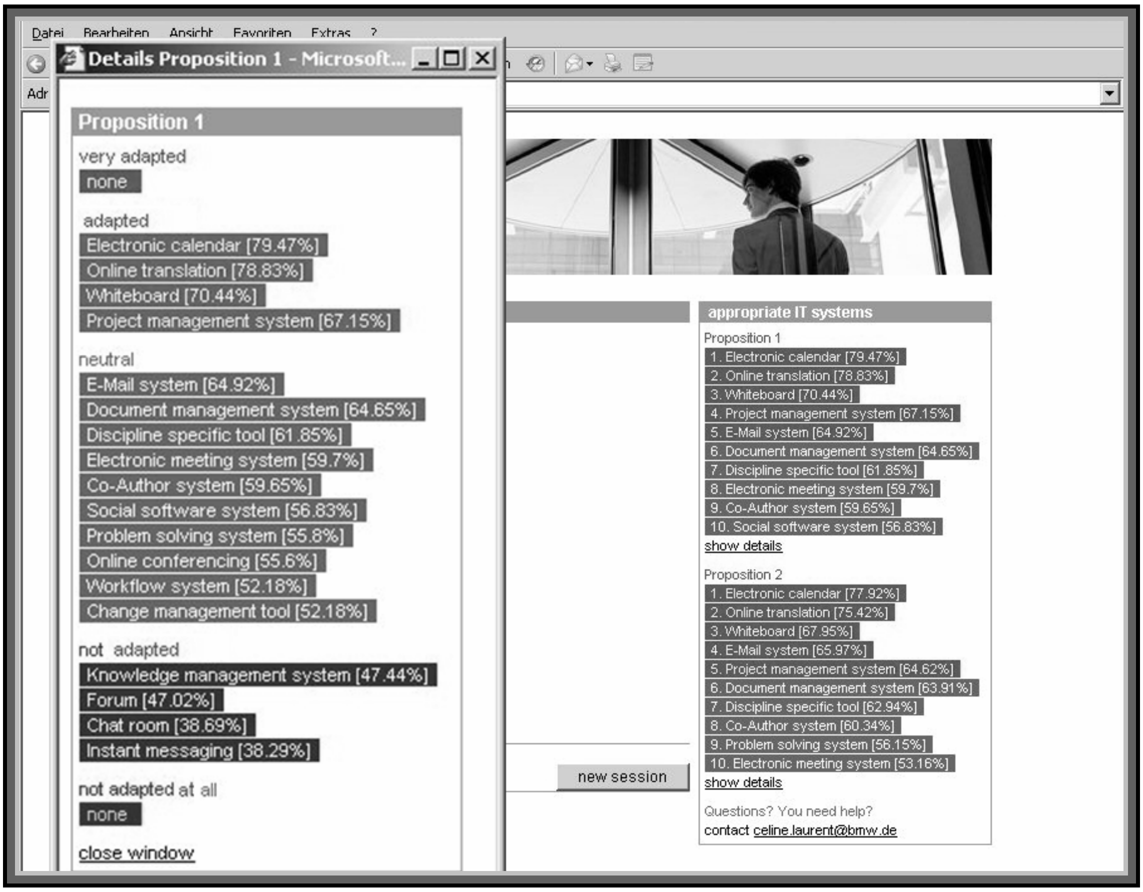
The vocabulary employed by the participants is different, since they are all specialists in different domains. They also use different methods and instruments depending on their respective domain and their mother company. However, they follow the same industrial standards. Otherwise it would be impossible to produce an engine. The dependence of the participants on their original company is very high due to the organisations' tight culture.

We now turn to the method described in Chapter 4 to propose an appropriate configuration of IT-platform. Toward this end we use the CPC described in 4.6. Figure 4-44 shows the input of this project, and respectively Figure 4-45 the propositions made by the CPC.



N°	Factor	Input
1	**Number of cooperation partners**	2 partners
2	**Location**	Other country
3	**Skill level in the agreed language**	60%
4	**Type of engineering process**	9/10 tasks
5	**Organisation and company's culture**	Other company
6	**Size of the organisation**	All > 1100 employees
7	**Intensity of collaboration**	50%
8	**Distribution model**	50/50%
9	**Number of interfaces**	12
10	**Access to data**	1/5 (0=very difficult)
11	**Skill level in the use of IT**	20%
12	**Influence of time**	70% (time remaining)
13	**Methods and instruments**	40%
14	**Vocabulary**	40%
15	**Standards and laws**	90%
16	**Dependencies on own domain**	70%

Figure 4-44: Input corresponding to the engine-development.


Datei Bearbeiten Ansicht Favoriten Extras ?
Details Proposition 1 - Microsoft...
Adr
new session

Proposition 1

very adapted
none

adapted
Electronic calendar [79.47%]
Online translation [78.83%]
Whiteboard [70.44%]
Project management system [67.15%]

neutral
E-Mail system [64.92%]
Document management system [64.65%]
Discipline specific tool [61.85%]
Electronic meeting system [59.7%]
Co-Author system [59.65%]
Social software system [56.83%]
Problem solving system [55.8%]
Online conferencing [55.6%]
Workflow system [52.18%]
Change management tool [52.18%]

not adapted
Knowledge management system [47.44%]
Forum [47.02%]
Chat room [38.69%]
Instant messaging [38.29%]

not adapted at all
none

close window

appropriate IT systems

Proposition 1
1. Electronic calendar [79.47%]
2. Online translation [78.83%]
3. Whiteboard [70.44%]
4. Project management system [67.15%]
5. E-Mail system [64.92%]
6. Document management system [64.65%]
7. Discipline specific tool [61.85%]
8. Electronic meeting system [59.7%]
9. Co-Author system [59.65%]
10. Social software system [56.83%]
show details

Proposition 2
1. Electronic calendar [77.92%]
2. Online translation [75.42%]
3. Whiteboard [67.95%]
4. E-Mail system [65.97%]
5. Project management system [64.62%]
6. Document management system [63.91%]
7. Discipline specific tool [62.94%]
8. Co-Author system [60.34%]
9. Problem solving system [56.15%]
10. Electronic meeting system [53.16%]
show details

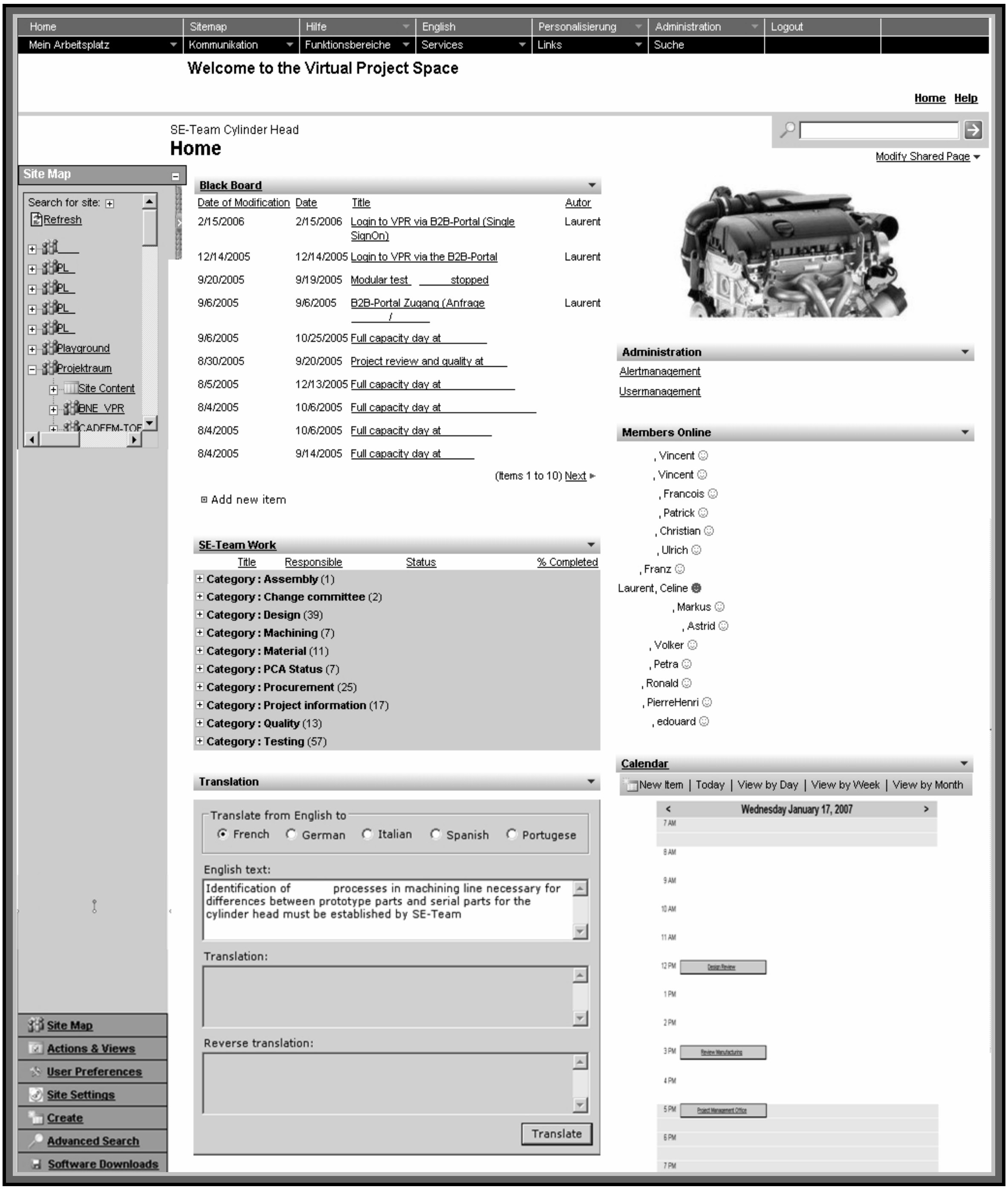
Questions? You need help?
contact celine.laurent@bmw.de

Figure 4-45: Proposed configuration for the engine-development project.

Realisation

The proposition has been used to build an IT-platform based on the Virtual Project Space (VPS) already introduced in section 4.7.1.1.

We have configured the VPS according to Figure 4-45 with an electronic calendar, a project management system, a whiteboard and an online-translator. E-Mails are still appropriate to support the communication of this project and could also be integrated into the platform through the ‘My Inbox Web Part’, so that every participant has its Project-E-Mails centralized on the platform (cf. [Microsoft, 2004]). Figure 4-46 presents the resulting collaboration-platform.



Welcome to the Virtual Project Space

SE-Team Cylinder Head

Home

Black Board

Date of Modification	Date	Title	Autor
2/15/2006	2/15/2006	Login to VPR via B2B-Portal (Single SignOn)	Laurent
12/14/2005	12/14/2005	Login to VPR via the B2B-Portal	Laurent
9/20/2005	9/19/2005	Modular test stopped	
9/6/2005	9/6/2005	B2B-Portal Zugang (Anfrage /	Laurent
9/6/2005	10/25/2005	Full capacity day at	
8/30/2005	9/20/2005	Project review and quality at	
8/5/2005	12/13/2005	Full capacity day at	
8/4/2005	10/6/2005	Full capacity day at	
8/4/2005	10/6/2005	Full capacity day at	
8/4/2005	9/14/2005	Full capacity day at	

(Items 1 to 10) Next ►

⊞ Add new item

SE-Team Work

Title	Responsible	Status	% Completed

- ⊞ **Category : Assembly** (1)
- ⊞ **Category : Change committee** (2)
- ⊞ **Category : Design** (39)
- ⊞ **Category : Machining** (7)
- ⊞ **Category : Material** (11)
- ⊞ **Category : PCA Status** (7)
- ⊞ **Category : Procurement** (25)
- ⊞ **Category : Project information** (17)
- ⊞ **Category : Quality** (13)
- ⊞ **Category : Testing** (57)

Translation

Translate from English to

● French ○ German ○ Italian ○ Spanish ○ Portugese

English text:

Identification of processes in machining line necessary for differences between prototype parts and serial parts for the cylinder head must be established by SE-Team

Translation:

Reverse translation:

Translate

Administration

Alertmanagement

Usermanagement

Members Online

, Vincent
, Vincent
, Francois
, Patrick
, Christian
, Ulrich
, Franz
Laurent, Celine
, Markus
, Astrid
, Volker
, Petra
, Ronald
, PierreHenri
, edouard

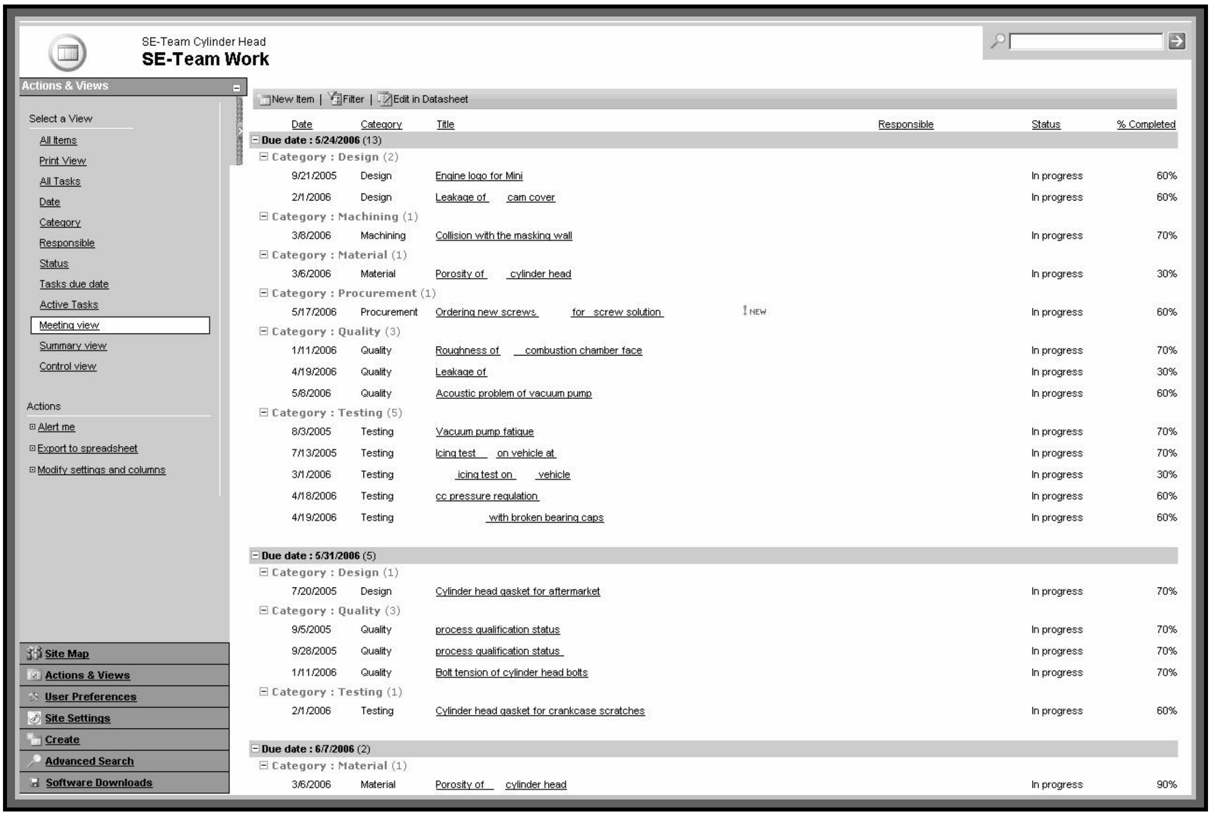
Calendar

Figure 4-46: Proposed collaboration-platform for the engine-development project.

The main focus of the platform is the project management system. In fact, it facilitates the organisation and efficiency of meetings, the overview and archiving of open points, creation of protocols, control of progressing, etc.

Prior to a meeting, every project member must fill out the points he is responsible for. Toward this end he has a personified view of his tasks. During meetings a special view of the open points is available to follow-up the various actual topics. Another view summarizes the different points discussed; it contains the subject of each point, the description of the events done last week, the individual responsible and the date of the next info. This summary view permits a quick printing of the topics discussed during meetings and so enables a rapid creation of protocols.

This intelligent list also follows the history of each point or task. Thus, at a single glance you can have a quick and complete overview of each task and open point.



Date	Category	Title	Responsible	Status	% Completed
Due date : 5/24/2006 (13)					
Category : Design (2)					
9/21/2005	Design	Engine logo for Mini		In progress	60%
2/1/2006	Design	Leakage of cam cover		In progress	60%
Category : Machining (1)					
3/8/2006	Machining	Collision with the masking wall		In progress	70%
Category : Material (1)					
3/6/2006	Material	Porosity of cylinder head		In progress	30%
Category : Procurement (1)					
5/17/2006	Procurement	Ordering new screws for screw solution ! NEW		In progress	60%
Category : Quality (3)					
1/11/2006	Quality	Roughness of combustion chamber face		In progress	70%
4/19/2006	Quality	Leakage of		In progress	30%
5/8/2006	Quality	Acoustic problem of vacuum pump		In progress	60%
Category : Testing (5)					
8/3/2005	Testing	Vacuum pump fatigue		In progress	70%
7/13/2005	Testing	Icing test on vehicle at		In progress	70%
3/1/2006	Testing	icing test on vehicle		In progress	30%
4/18/2006	Testing	cc pressure regulation		In progress	60%
4/19/2006	Testing	with broken bearing caps		In progress	60%
Due date : 5/31/2006 (5)					
Category : Design (1)					
7/20/2005	Design	Cylinder head gasket for aftermarket		In progress	70%
Category : Quality (3)					
9/5/2005	Quality	process qualification status		In progress	70%
9/28/2005	Quality	process qualification status		In progress	70%
1/11/2006	Quality	Bolt tension of cylinder head bolts		In progress	70%
Category : Testing (1)					
2/1/2006	Testing	Cylinder head gasket for crankcase scratches		In progress	60%
Due date : 6/7/2006 (2)					
Category : Material (1)					
3/6/2006	Material	Porosity of cylinder head		In progress	90%

Figure 4-47: Web Part for project management.

The whiteboard (called ‘Black Board’ on Figure 4-46) is the first element you see when you enter the platform. It contains information given by the SE-Team members shown directly on the top of the homepage. The calendar shows all of project’s relevant dates, deadlines and meetings. To enable online-translation we could integrate, for example, a tool like the Google Translator Web Part visible on Figure 4-46 (cf. [Google, 2007] and [Code Zone, 2005]).

4.7.2. Application to a Software Engineering Project

Description of the project

With a presence in 190 countries and over 30,000 software developers, the percentage of Siemens projects that are globally distributed has been increasing steadily. Siemens Corporate Research, Inc. (SCR) has been doing research aimed at developing a better understanding of the issues and impact of various practices with respect to Global Software Development (GSD). One of the initiatives and part of this research is the Global Studio Project (GSP), which has organized the work of Software Engineering and Computer Science student teams from five universities in four countries into a single global project. Here, we present and analyze experiences made by teams from three universities (cf. [Keil et al., 2006]).

The universities included are Carnegie Mellon University, Monmouth University (both USA), University of Limerick (Ireland), iiit-b (India), and Technische Universität München (TUM, Germany) (cf. [Hummel et al., 2005]). To get results closely related to problems appearing in real projects, the GSP was set up to precede an actual GSD project at Siemens. GSP was planned to start half a year before the real project at Siemens to anticipate possible problems that might occur for the larger project. More concretely, the goal was to test the suitability of the process model and the tool chain for globally distributed development and to learn more about the communication patterns used between the local teams and the central team at SCR (especially since the locations were in different time zones). The product developed in the 'real' project is a management station for buildings. To make the simulated project as similar as possible to that, the product to be developed is a simplified management station working on top of a 'simulated building'. This system is called Management Station Lite or MSLite for short. To make it more manageable MSLite is divided into three major blocks (cf. [Hummel et al., 2005]):

- Field System Simulator (FSS): the FSS is the simulated building. Similar to a real building it sends information concerning the change of the state of its components (e.g. temperature sensors, smoke detectors, elevators).
- Core: this is the server part of MSLite. It is built around the System Object Model (SOM) which represents the current state of the field systems (building) and a Logic-And-Reaction (L+R) component used to perform actions in response to events (e.g. raise fire alarm, if smoke detector is activated).
- Presentation: the presentation system handles user interaction by utilizing the services provided by the core system. As it is based on ASP.NET most of it runs on top of a web server and multiple clients can connect using a standard web browser.

This experiment permitted the investigation of particular questions to an extent that would not be possible in any large industrial project. The five universities set up seven distributed student development teams that were responsible for design, development and unit test for defined work packages (core modules / sub-systems). The SCR central team was responsible for the high-level requirements, software architecture, system test and integration, and project management.

Supplier managers managed particular student teams. They were responsible for monitoring the progress of the team, being the first point of contact and coordinating the communication between the teams and other project members. Teams, as a rule, were not permitted to communicate directly with one another but spoke directly to their supplier manager.

The project plan reflected the iterative nature of the intended development process. Due to the communication issues typically experienced in GSD projects, additional controls were put in place to track progress against the plan and to identify issues early. One of the approaches taken to accomplish this was to define four- to six-week milestones with clearly defined deliverables by all the teams and an integration task at each of the milestones (engineering releases). This project already has some collaboration-tools in use, but not in form of a collaboration-platform.

Application of the method

First, we describe the project with respect to the influencing factors described in sections 2.4 and 4.3.1. Toward this end we will also use the set of natural-language words mentioned in the previous chapter.

The project comprises more than five cooperation partners, which are located in various countries. The partners are from different companies, or in this case universities with different organisations' culture divided into two categories: large-scale enterprises and middle-scale organisations. The global skill level of the participants in the project language is average, but does not lead to considerable contradictions.

Generally, the participants work parallelly on their work packages. However, the collaboration is quite low and irregular, probably because the teams, as a rule, were not permitted to communicate directly with one another. The tasks are more or less equally distributed between the different teams. Only the central team does more than the other teams, approximately a third of the entire work.

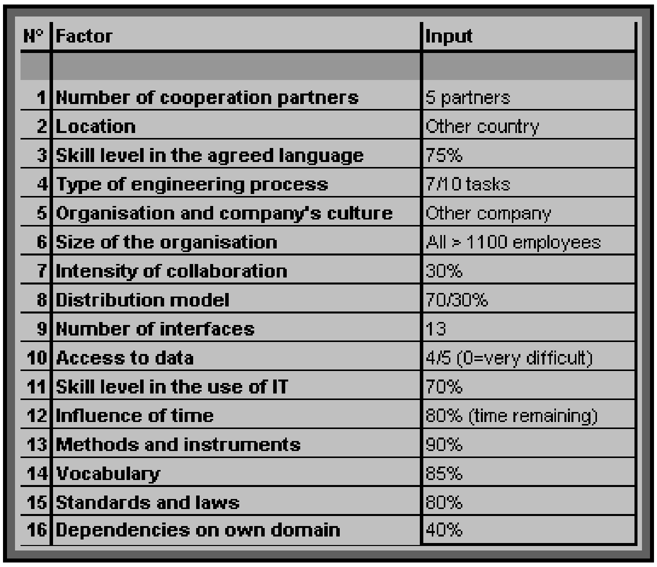
There are very few technical interfaces; nevertheless organisational interfaces are quite striking, especially because the teams are not allowed to communicate together. The technical and organisational ability of every partner to access the relevant data is quite good. GSP is not in a security-sensitive work context.

The skill level of the participants in the use of IT is quite good. Although it is not their speciality, in the majority of cases they have IT background and can easily deal with it.

This project is a continuous and long-lasting project and hence not really time-critical.

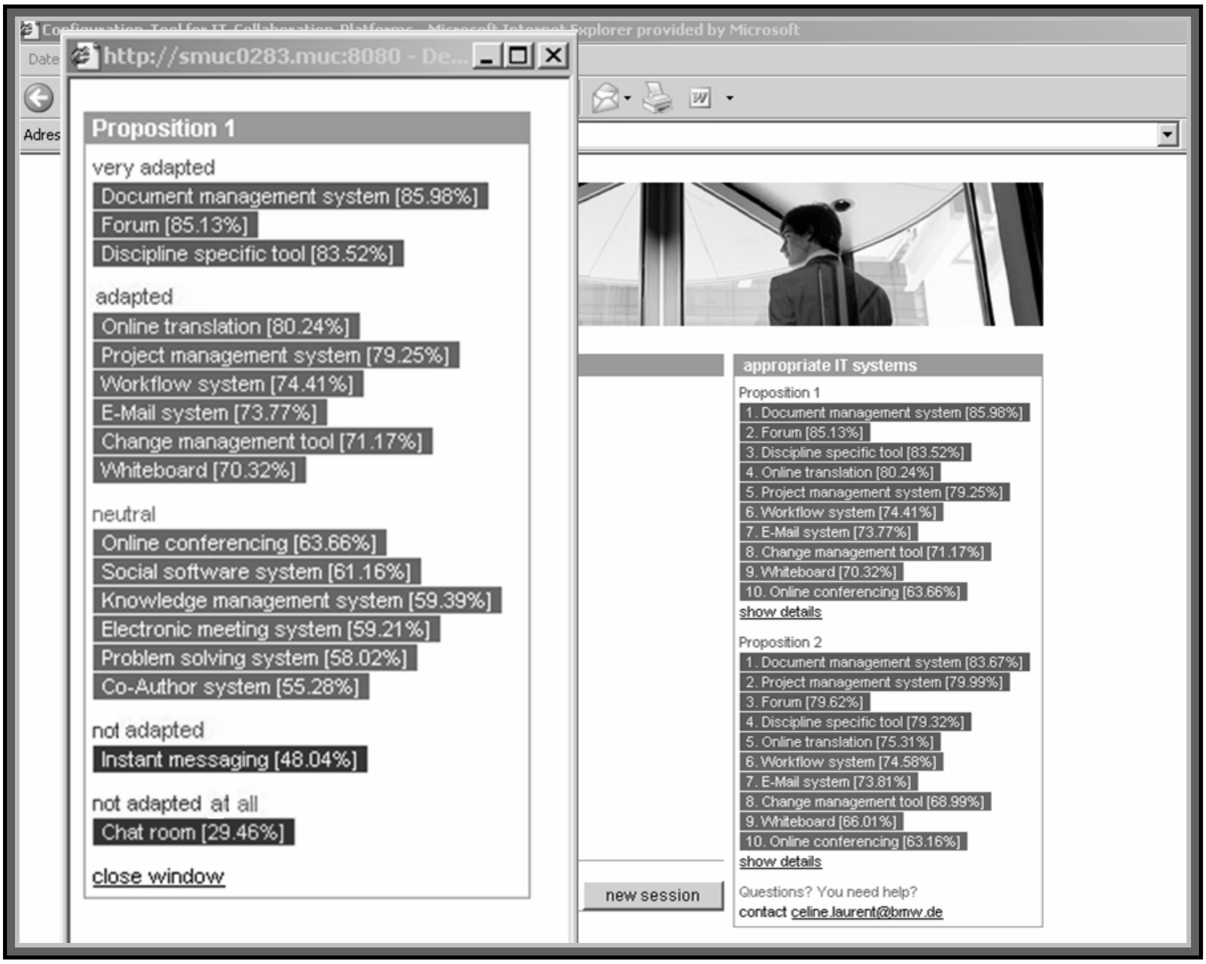
The vocabularies employed by the participants are for the most part the same. They are all specialists in the same fields of works. They also use on the whole the same methods and instruments and follow approximately the same standards imposed by the central team. Dependence on their original organisation is perceptible, resulting probably from the lack of meetings with physical presence. They work first for their organisation before working for the project.

Using the method described in Chapter 4 we now propose an appropriate configuration of IT-platform. Toward this end we use the CPC described in 4.4.3. Figure 4-48 shows the input of this project, and respectively Figure 4-49 the propositions made by the CPC.



N°	Factor	Input
1	Number of cooperation partners	5 partners
2	Location	Other country
3	Skill level in the agreed language	75%
4	Type of engineering process	7/10 tasks
5	Organisation and company's culture	Other company
6	Size of the organisation	All > 1100 employees
7	Intensity of collaboration	30%
8	Distribution model	70/30%
9	Number of interfaces	13
10	Access to data	4/5 (0=very difficult)
11	Skill level in the use of IT	70%
12	Influence of time	80% (time remaining)
13	Methods and instruments	90%
14	Vocabulary	85%
15	Standards and laws	80%
16	Dependencies on own domain	40%

Figure 4-48: Input corresponding to GSP.


xplorer provided by Microsoft
Date
Adres
http://smuc0283.muc:8080 - De...
new session

Proposition 1

very adapted

- Document management system [85.98%]
- Forum [85.13%]
- Discipline specific tool [83.52%]

adapted

- Online translation [80.24%]
- Project management system [79.25%]
- Workflow system [74.41%]
- E-Mail system [73.77%]
- Change management tool [71.17%]
- Whiteboard [70.32%]

neutral

- Online conferencing [63.66%]
- Social software system [61.16%]
- Knowledge management system [59.39%]
- Electronic meeting system [59.21%]
- Problem solving system [58.02%]
- Co-Author system [55.28%]

not adapted

- Instant messaging [48.04%]

not adapted at all

- Chat room [29.46%]

close window

appropriate IT systems

Proposition 1

1. Document management system [85.98%]
2. Forum [85.13%]
3. Discipline specific tool [83.52%]
4. Online translation [80.24%]
5. Project management system [79.25%]
6. Workflow system [74.41%]
7. E-Mail system [73.77%]
8. Change management tool [71.17%]
9. Whiteboard [70.32%]
10. Online conferencing [63.66%]

show details

Proposition 2

1. Document management system [83.67%]
2. Project management system [79.99%]
3. Forum [79.62%]
4. Discipline specific tool [79.32%]
5. Online translation [75.31%]
6. Workflow system [74.58%]
7. E-Mail system [73.81%]
8. Change management tool [68.99%]
9. Whiteboard [66.01%]
10. Online conferencing [63.16%]

show details

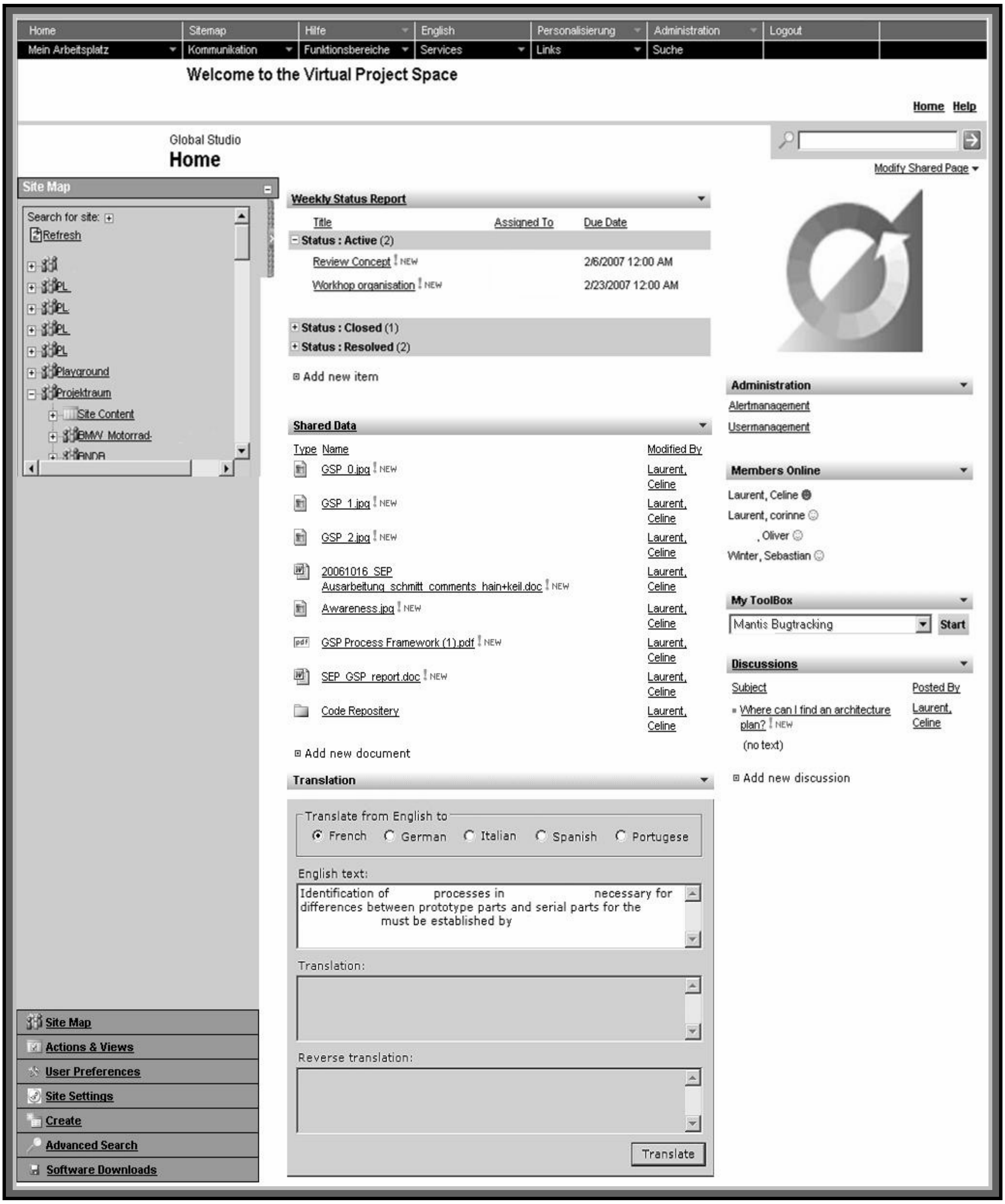
Questions? You need help?
contact celine.laurent@bmw.de

Figure 4-49: Proposed configuration for GSP.

Realisation

The proposition has been used to build an IT-platform based on the Virtual Project Space (VPS) already introduced in section 4.7.1.1.

We have configured the VPS as seen in Figure 4-49 with a document management system, a forum, a discipline specific tool, an online-translator, and a project management system. It is interesting to note that E-Mails are not appropriate to support the communication on this project, despite the fact that E-mail earns its role as a major channel for corporate information - both official and unofficial (cf. [Stenmark, 1999]). Figure 4-50 presents the resulting collaboration-platform.



Welcome to the Virtual Project Space

Global Studio

Home

Weekly Status Report

Title	Assigned To	Due Date
Status : Active (2)		
Review Concept ! NEW		2/6/2007 12:00 AM
Workhop organisation ! NEW		2/23/2007 12:00 AM
Status : Closed (1)		
Status : Resolved (2)		

Add new item

Shared Data

Type	Name	Modified By
	GSP_0.jpg ! NEW	Laurent, Celine
	GSP_1.jpg ! NEW	Laurent, Celine
	GSP_2.jpg ! NEW	Laurent, Celine
	20061016_SEP Ausarbeitung_schmitt_comments_hain+keil.doc ! NEW	Laurent, Celine
	Awareness.jpg ! NEW	Laurent, Celine
pdf	GSP Process Framework (1).pdf ! NEW	Laurent, Celine
	SEP_GSP_report.doc ! NEW	Laurent, Celine
	Code Repository	Laurent, Celine

Add new document

Translation

Translate from English to

● French ○ German ○ Italian ○ Spanish ○ Portugese

English text:

Identification of processes in necessary for differences between prototype parts and serial parts for the must be established by

Translation:

Reverse translation:

Administration

Alertmanagement

Usermanagement

Members Online

Laurent, Celine ●

Laurent, corinne ○

, Oliver ○

Winter, Sebastian ○

My ToolBox

Mantis Bugtracking

Discussions

Subject	Posted By
Where can I find an architecture plan? ! NEW (no text)	Laurent, Celine

Add new discussion

Figure 4-50: Proposed collaboration-platform for GSP.

In this case the discipline specific tool is a toolbox permitting direct access to the development tools used in the project like Mantis Bug-Tracking or Cruise Control. Since they are not standard tools, they are not automatically available in VPS. The toolbox enables this direct access in the project context. Forum, project management system, document management system and online-translator are the standard tools of such platforms. The document management system of VPS with check-in/out functions is used as a code and document repository of the project, and hence contains all specifications, results of integration and acceptance test, and so on. The project management system includes project planning, the various agenda and minutes of meetings, as well as a weekly status report on the current project status (products, understanding of requirement, etc.) and communicational aspects (number of emails sent, number of phone calls, etc.). The forum acts as a communication tool between the different teams to solve problems or exchange information.

4.7.3. Interpretation of the Results

We have configured three collaboration-platforms with the help of the CPC. The use of CPC is quite easy and fast, and although knowledge of the project under consideration might be necessary, it does not require any special competence. In addition, we have implemented the results in form of a collaboration-platform based on the VPS in order to evaluate the solution proposed.

The selected projects display different aspects of the CPC. The two engineering projects in the automotive field show, first, the advantages of an optimal IT- support. The collaboration-platform has been configured and optimised to respond to the project's needs, and hence facilitates obtaining a quick overview of the project issues, centralises all up-to-date documents and project information, enables the generation of workflows, and permits the follow-up of each task and responsibility. The CPC and the modular configuration enable an individual design of the platform, which increases its acceptance by the participants, and thus increases the efficiency of the project work. Moreover, the integration of discipline-specific tools like the test module presented in paragraph 4.7.1.1 solves project-specific problems; in this case, it reduces travel costs and time involved in the development. Subsequently, the CPC and the modular configuration increase the project's efficiency due to a higher acceptance and appropriate IT-solutions.

The case of the software engineering project evidences differences, since it already has IT-collaboration-tools in use. However, they are not centralised in form of a platform. They used a co-author system as document repository, another IT-system as code repository, a forum, and other discipline-specific development tools (like “Mantis Bug-Tracking” or “Cruise Control”). Reports on this project state the following problems of the used IT-support:

- There was no tool for an automatic updating of the co-author system, i.e. every time an interface was changed or the architecture was enhanced, several co-author system pages needed to be updated. In addition, automatic checking for the conformance of documentation was not given.

- Too many emails and too sporadic use of forum and “Mantis Bug-Tracking”.
- Checking-in of test code at Cruise Control integration server costs a lot of time.
- Problems in the update mechanisms.

The proposition made by the CPC is quite similar to the IT-tools chosen for this project. However, the centralisation of every IT-tool and information on the platform is certainly an advantage, since it guarantees an automatic update of the information and quick access to every tool and repository. The check-in functions of the document management system also avoid some problems of the update mechanisms. The system has also discouraged the use of E-Mails as a communication tool in this project, which is one of the problems encountered during the project.

This project displays another aspect of this methodology. In fact, only with the CPC were we able to configure a collaboration-platform for this project. The solution proposed corresponds actually to the project’s needs and has been constructed without meetings, brainstorming, consulting, or time-consuming analysis of IT-tools, or of the project, which would also have permitted a quick estimation of the IT-costs at an early stage of the project. Moreover, the configuration of the platform didn’t take more than one hour.

Furthermore, it is important to note that we have used the same platform for each project and each configuration. Configuring a standard tool facilitates a quick implementation and avoids the costs of new software development.

5. EVALUATION OF THE METHODOLOGY

The case studies presented in section 4.7 provide a means to evaluate the method of configuring IT-collaboration-platforms. Implementing the configured platform also highlights the limitations of this method and indicates which aspects must be kept in mind during the procedure. This chapter presents the limitations of the applicability of this method as well as an economic assessment of its benefits.

5.1. Applicability and Limitations of the Method

The method to configure IT-collaboration-platforms is in the whole intended for engineering projects, especially engineering projects with cross-enterprise and cross-domain characteristics. In fact, this method helps projects and IT-managers in decision-making regarding the right IT-collaboration-platforms for CEE and CDE projects. This aims at reducing the development and implementation time of the platform within an engineering project.

The method is based on an empirical study that analyses the correspondence between IT-tools and CDE/CEE influencing factors. The method is not specific to one engineering field. In fact, we have tested it not only for the same kinds of engineering processes, but also for different processes and products (e.g. engine-development and software engineering project). The projects under consideration are very different, involving various cultures, organisations, and disciplines. Thus, the method is easily applicable to various kinds of engineering projects. However, to use it for other projects, such as a marketing or a financial project, requires a review of the factors. Factors specific to engineering projects would be omitted, while other factors must be included.

The limitations of the method can be classified into three different categories:

- limitations of the method and CPC: “where does the method help, where not?”
- risks of considering the single evaluation by the project manager
- problems of using an empirical basis

The method gives the project leader an advice on how his collaboration-platform has to be configured, i.e. which IT-tools are appropriate or not for his project. It is important to note that even if the collaboration platform is very efficient and corresponds to the project’s needs, ultimately its success depends on the users, their acceptance, the manner of introducing the new tool and, most importantly, the new processes to be followed. The developed methodology neither provides advice or guidelines for implementing the proposed solution, nor an explanation why an IT-tool is recommended or not. Therefore, it is not evident that the users understand these recommendations or accept them readily. The team leader should take the suggestions made by the CPC into account and configure an appropriate platform, as exemplified in the case studies, before presenting it to the team.

Secondly, there is also a certain risk in using the CPC, since the results depend on the evaluation or estimation of the project leader only, which can potentially distort the recommendation. A proposition is made in section 6.2.1 to minimise this risk.

Finally, the method is based on an empirical study, which also may be questionnable. E.g. the assumption that there is in principle a unique and neutral empirical basis has been seriously questioned in, for example, [Lakatos, 1970] and [Weinberg, 1960]. However, it is the essence of an expert-system to use human knowledge, and thus empirical data, to solve problems that normally would require human intelligence. In this case, data and rules given by experts in the domain considered are normally used as knowledge-base. Consequently, an expert-system does not guarantee a unique true solution, but acts as a human-expert.

Another problem of the empirical basis is that each improvement of the expert-system, for example the inclusion of new IT-tools, would require new obligatory empirical research, since the assessment of new IT-tools by experts has to be added to the knowledge-base. This may be a critical point since technology is always in evolution and the number of IT-tools continually increasing.

5.2. Economical Aspects

As mentioned in section 2.2, the fundamental success factors of an enterprise to meet its objectives and maximise the performance are quality, costs, and the time to market (cf. [Danner, 1996]). In the case studies discussed in section 4.7, some aspects of the developed methodology emerge, which have a definite influence on these success factors. We comment on these aspects in the following three aspects:

- What are the benefits of using a standard platform? (see §5.2.1)
- What are the benefits of using an individually-tailored collaboration-platform based on a standard platform? (see §5.2.2)
- What are the benefits of using the methodology to configure a platform? (see §5.2.3)

5.2.1. Benefits of a Standard Platform

In this section we show the benefits of using a standard tool to configure our collaboration-platforms. In fact, we could have developed a new collaborative software with the recommendations of the CPC each time, but we chose instead to configure repeatedly the same platform: VPS. Of course, it is only possible with a platform that is based on a flexible architecture which permits a free configuration of functionalities and tools. It probably makes a lot of savings possible.

First, using a standard platform reduces the economic risk of implementing new software. In fact, standard-tools are not developed to satisfy one project's objective. Thus, if it is a failure in one project, the time and money invested are not lost, since the tool is used in other projects.

Standards do not only reduce the economic risk of development activities. According to [DIN, 2000] they can also lower the enterprise's development costs. The expense of development activities can be reduced when the participants in standards make their results generally available, avoiding duplications. In our case we use a standard tool instead of developing new software, thereby completely eliminating the development costs and enabling a very fast implementation of the tool. This detail is very important, if considering the time necessary to construct an operational system. The training and configuration costs, however, can not be omitted.

Using standards also helps to ensure that everyone can have access to the platform provided. It also contributes to lower accident rates or failures. In comparison to new software a standard tool has processes established, as well as maintenance, support processes, material, and documentation. Indirectly it also augments the life cycle of the tool, since it is used more often and accepted as a standard.

5.2.2. Benefits of Individually-Tailored IT-Collaboration-Platforms

In this section we demonstrate the benefits of using an optimal and individual collaboration-platform adapted to specific project configurations.

The platforms configured with the help of the methodology take into account particular characteristics of development projects and human factors. These factors are very difficult to quantify, hence the use of fuzzy logic, which considers human preferences and so guarantees a high user acceptance. The resulting collaboration-platforms are optimised to the projects' needs and follow the current business and engineering processes. The CPC and the modular configuration facilitate an individual design of the platform, which increases its acceptance by the participants, and thus increases the efficiency of the project work.

First, of course, in rank is the criterion of quality. In fact, the optimised platforms are adequate to the project configuration and thus clarify and structure the collaboration processes, such as information exchange, document management, coordination, planning, and so on. But does the use of an optimal collaboration-platform have repercussions on the enterprise's benefits?

When adopting a software technology, like the generated IT-collaboration-platforms, companies look for clear-cut benefits that have a measurable impact on their work processes and business results. Often these benefits are calculated with the help of a return-on-investment (ROI) analysis. It is the ratio of money gained or lost on an investment relative to the amount of money invested (cf. [Pau, 2002]):

ROI= (net benefice p.a./invest)x100, where:

- Net benefice p.a.= benefits p.a. – (current costs after launch p.a. + (one-off costs of implementation / service life of the application)
- Invest = one-off costs of implementation

The goal of a ROI Model is to establish the value of a project by calculating its expected return. A return-on-investment model can be developed by looking at all the inputs and assumptions that go into the ROI equation. To start with, one must look at the two main inputs in the ROI calculation: benefits and associated costs. The application of a tailored IT-collaboration-platform influences the benefits of projects and thus influences the global ROI calculation.

These benefits are often divided into direct and indirect benefits. Direct benefits result in a tangible cash benefit. Strong ROI models typically derive the majority of their benefits from hard-dollar savings. Indirect benefits are not easily quantifiable in hard-dollar terms. Although they do not provide a cash benefit, indirect benefits are important since they quantify other factors that may be important when evaluating whether to proceed with the investment or not. Depending on the project, the benefits of using a collaboration-platform can vary. However, it is evident, that collaboration-tools drastically increase the ROI (cf. [Okujava, 2006]). Table 5-1 shows typical benefits of using such a collaboration-platform in a project:

Software type	Sample benefit	Benefit classification	Benefit description
Collabora-tion - platforms	Reduce or eliminate travel, reap significant cost savings	Direct	On site meetings are easily replaced by a web conference using the platform to set up and move through a structured agenda of discussion topics.

	Increase productivity, produce faster results	Direct	Collaboration tools increase meeting productivity. A focused simultaneous brainstorming reduces what can take up to hours in a traditional setting to merely minutes. Similarly, categorizing and voting quickly narrow the field of ideas to the ones that will really make a difference. The time saved can be redeployed into taking action or reinvested into taking the discussion to the next level.
	Increase participation	Indirect	It increases the collaboration by making it easy to get input from a broader spectrum of customers, employees, suppliers and stakeholders with any time any place surveys and conferences. The results: decisions are made with a higher degree of consensus and agreements are more binding.
	Productivity savings	Indirect	Value of reduced effort spent on the process, which can not be tied directly to cash results.
	Improved quality	Indirect	An intangible measure of product, customer service, or operational effectiveness that is often difficult to tie to cash results.
	Eliminate paper printings	Direct	It facilitates creating questionnaires, tasks lists, forums, etc. without programming; it notifies participants by email and tabulates results online instantly. With no printing or mailing, no data entry or analysis costs, it drastically reduces the enterprise’s expenses.
	Improved information	Direct or indirect	Improved decision-making that results from having access to timelier and/or more accurate information which leads to improved business results or productivity gains.

Table 5-1: Examples of collaboration benefits in a ROI analysis.

The ROI calculation would then have the following elements (see equation before):

- Benefits p.a.: see Table 5-1
- Current costs after launch p.a.: maintenance, support, software upgrades
- One-off costs of implementation: implementation, configuration with the help of the CPC, training, software licences (no development costs since we use a standard platform)
- Service life of the application: depending on the context or enterprise (at BMW normally from 5 to 10 years)

Considering a single project, the implementation costs could be drastically reduced, since we must divide them into the number of projects using the standard platform. This ROI analysis aids in understanding the benefits of implementing an individually-tailored collaboration-platform, especially if it is based on a standard platform.

5.2.3. Benefits of the Methodology

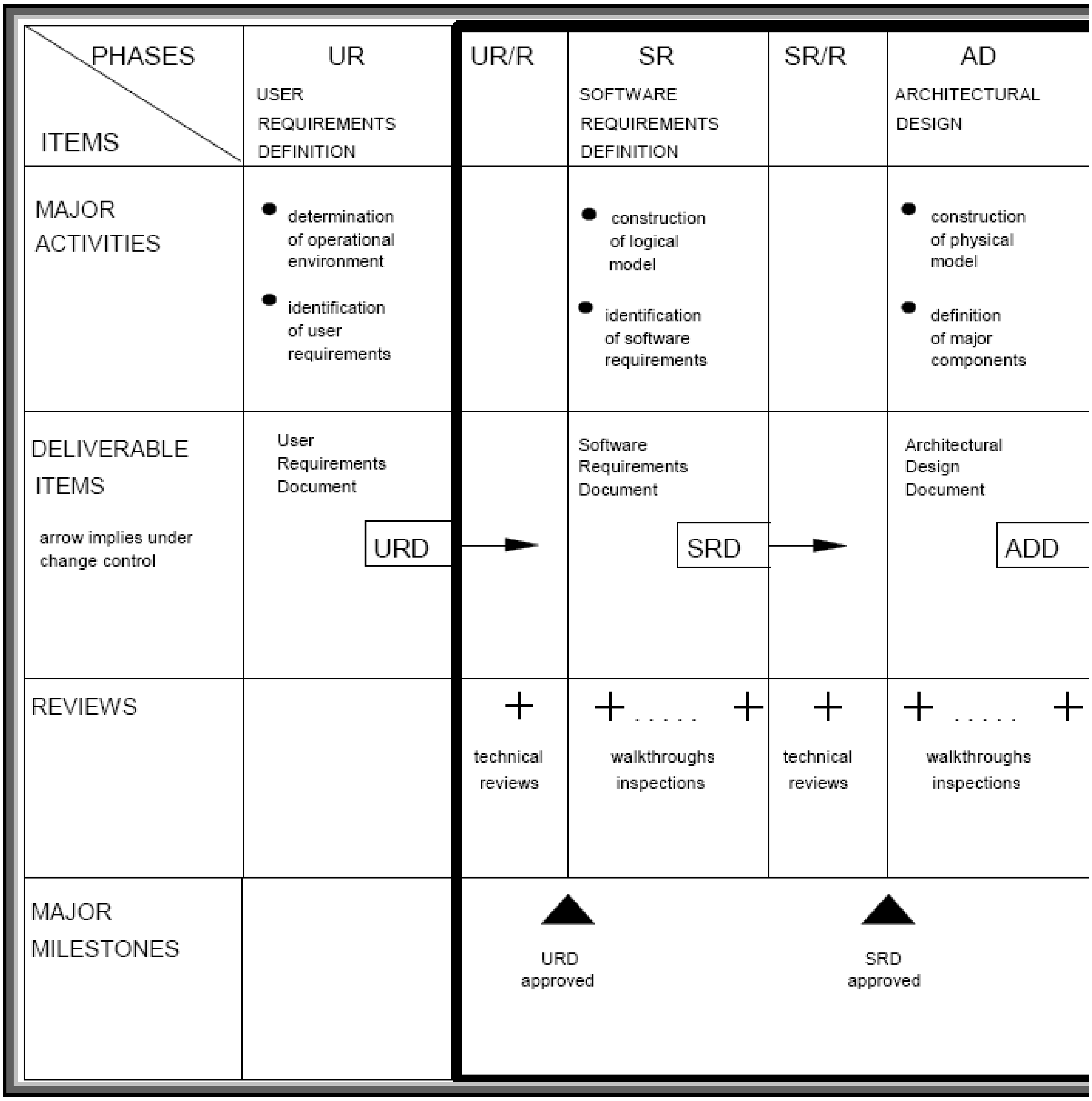
In this section we demonstrate that using the methodology and respectively the CPC reduces the 'time to application' of the collaboration-platform. We use the term 'time to application' and not 'time to market', since our goal is to create an operational system or application for collaboration, which means also that the user are trained and able to use it, and not a marketable product.

By basing our analysis on the software life cycle according to [ESA, 1991], we will demonstrate that using the method reduces the 'time to application'. A software life cycle starts when a software product is conceived and ends when it is no longer available for use. The products of a software development project should be delivered as quickly as possible and be suitable for their purpose. A 'life cycle model' structures project activities in 'phases' and defines which activities occur during which phase. Figure 5-1 shows the life cycle model used in these standards.

Software projects often have a life-cycle approach which normally includes the phases shown in Figure 5-1 and Figure 5-2:

- UR phase - Definition of the user requirements
- SR phase - Definition of the software requirements
- AD phase - Definition of the architectural design
- DD phase - Detailed design and production of the code
- TR phase - Transfer of the software to operations
- OM phase - Operations and maintenance

In the context of 'time to application' this approach of the software life cycle should be extended with a phase comprising user trainings as well as procedures to make him accept the software such as communication measures, consulting, and rollouts.



PHASES ITEMS	UR USER REQUIREMENTS DEFINITION	UR/R	SR SOFTWARE REQUIREMENTS DEFINITION	SR/R	AD ARCHITECTURAL DESIGN
MAJOR ACTIVITIES	● determination of operational environment ● identification of user requirements		● construction of logical model ● identification of software requirements		● construction of physical model ● definition of major components
DELIVERABLE ITEMS arrow implies under change control	User Requirements Document URD	→	Software Requirements Document SRD	→	Architectural Design Document ADD
REVIEWS		+ technical reviews	+ . . . . . + walkthroughs inspections	+ technical reviews	+ . . . . . + walkthroughs inspections
MAJOR MILESTONES		▲ URD approved		▲ SRD approved	

Figure 5-1: Software life cycle model, part 1(according to [ESA, 1991]).

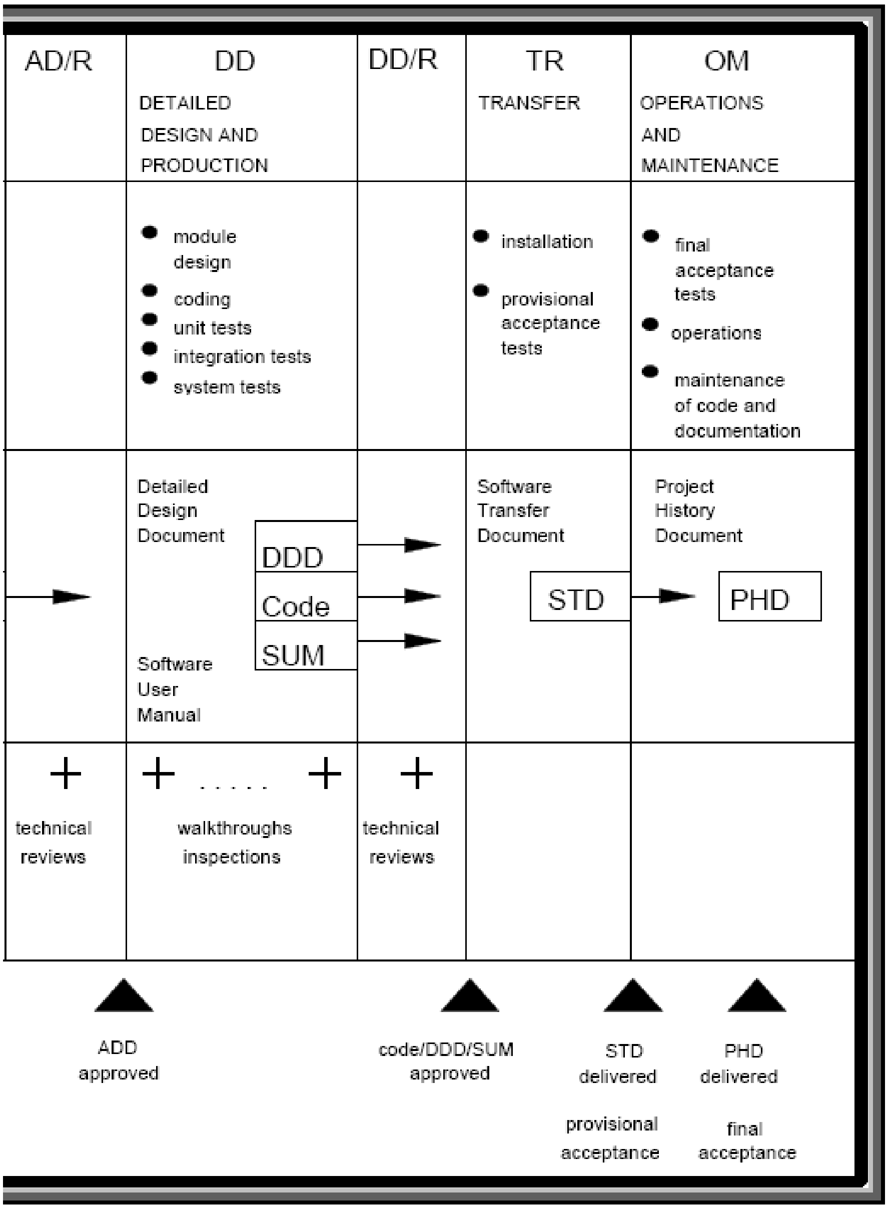


Figure 5-2: Software life cycle model, part 2(according to [ESA, 1991]).

Our method and the CPC aid project- and IT-managers in decision-making regarding the right IT-support-platform for CDE and CEE projects substantially and thus influences the phase, in which the user requirements are defined, i.e. the UR phase presented in Figure 5-1.

According to [ESA, 1991] the UR phase is the ‘problem definition phase’ of a software project. In this phase the scope of the software must be defined and user requirements captured, which is normally done by interview or survey, or by building prototypes. Specific user requirements must be identified and documented in the User Requirements Document (URD). The involvement of the developers in this phase varies according to the familiarity of the users with the software. Some users can produce a high-quality URD, while others may need help from the developers.

We will describe the activities carried out in this phase in detail in order to highlight the savings through the use of our methodology.

5.2.3.1. Capture of User Requirements

User requirements originate in the spontaneous perception of needs. However, these should be clarified through the criticism and experience of existing software and prototypes. The widest possible agreement about user requirements should be established through interviews and surveys. It also often happens in form of meetings or brainstorming. A user-requirements definition is an iterative process, and requirements-capture activities may have to be repeated several times before the URD is ready for review (cf. [ESA, 1991]).

The method developed aids the project leader in defining the functional user requirements. It is often the case for collaboration software that users want to collaborate, but do not know how. Moreover, as mentioned above the requirement-capture activity is iterative and must be repeated several times. By using the methodology and the CPC a number of time-consuming and expensive activities could be avoided.

First, meetings to define the scope of the software and the general requirements could be replaced by the results prevalent in the expert-system. Such meetings cost a great deal of time and money. Everyone must first travel to the conference site, which potentially can be far away, given CEE contexts, and thus generate travel and accommodation costs.

Moreover, some meetings seem to drag on as group members struggle to reach consensus and make decisions. By contrast, responsibility for making decisions rests squarely with the chair or leader.

Huge amounts of time and money are often wasted on trivia. The cost of a meeting could approximately be evaluated by multiplying the number of participants, their labour rate, and the length of the meeting. Then all other expenses could be added, which should include travel, materials, refreshments, room rental, and other expenses.

If we take into consideration that only two persons are necessary to define the global functional requirements and that they are located at a distance of 150km apart and have an hourly rate of approximately 100€ (cf. [AMT, 2007], [Gulp, 2007a], and [Gulp, 2007b]), initial costs would be:

- Travel costs: 150km by railway costs approximately 45€ in economy class, making a total of 90€ round-trip (cf. [DB, 2007]).
- Personal costs: a one-hour meeting costs 2 x 100€ = 200€ and 2 hours of travel (round-trip) for one of the one travelling (2 x 100€ = 200€)

If no expenses are incurred for room rental, material, and refreshments, the meeting will already cost 490€.

If more persons are necessary, perhaps a consultant or another specialist, possibly from another country, this would also generate accommodation costs and the costs of such a simple meeting augments drastically. By contrast, filling in the questionnaire of the CPC takes less than thirty minutes. Since the project leader is the only one to fill it in, and the system already provides him with a proposition of the software to develop, the costs would only amount to: ½ (h) x 100€ = 50 €.

Requirements are also commonly established through interviews and surveys. However, the interviews must be prepared, carried out and analysed. This process is very time-consuming. Even if an online-survey, which normally interprets the results automatically, is used, it takes a long time until anyone answers the questionnaire. Moreover, it implies expenses: an online-survey can cost up to 7000€ (cf. [Olss, 2000], [Smart-Research, 2007], and [SurveyPro, 2006]).

Thus, the minimum of savings by using the CPC in the functional user-requirement definition is:

$$\% \text{ savings} = 100- \frac{(\text{Costs of the meetings needed with CPC + Personal costs by using CPC})}{\text{Costs of the meetings needed without CPC}} \text{x}100$$

In our example, if the use of CPC can replace one of three necessary meetings, the % of savings will be:

%savings = 100 - ((490 x 2 + 50) / (490 x 3)) x 100 = 30 %

The more meetings, interviews, persons, surveys, and brainstorming you normally need to define the global functional user requirements and you can replace by using CPC, the merrier the benefits made in this phase are.

5.2.3.2. Determination of Operational Environment

According to [ESA, 1991] this step should be the first one in defining the user requirements. This narrative description gives a statement about the real world in which the software is to operate. However this description is not really necessary with the CPC, since it already takes into account the configuration in which the software is to operate. However, if they do exist, the nature of exchanges with external systems should be specified in this step and controlled from the start of the project. This narrative description may be supported by context diagrams, to summarise the interfaces with external systems, and system block diagrams to show the role of the software in a larger system (cf. [ESA, 1991]).

Once the operational environment has been established, specific user requirements are extracted and organised. Implementation considerations are omitted, unless they are the essence of the requirement. Specific user requirements are so far contained in the results of the CPC. However this step has to be carried out to review the solution of the CPC.

5.2.3.3. Outcome

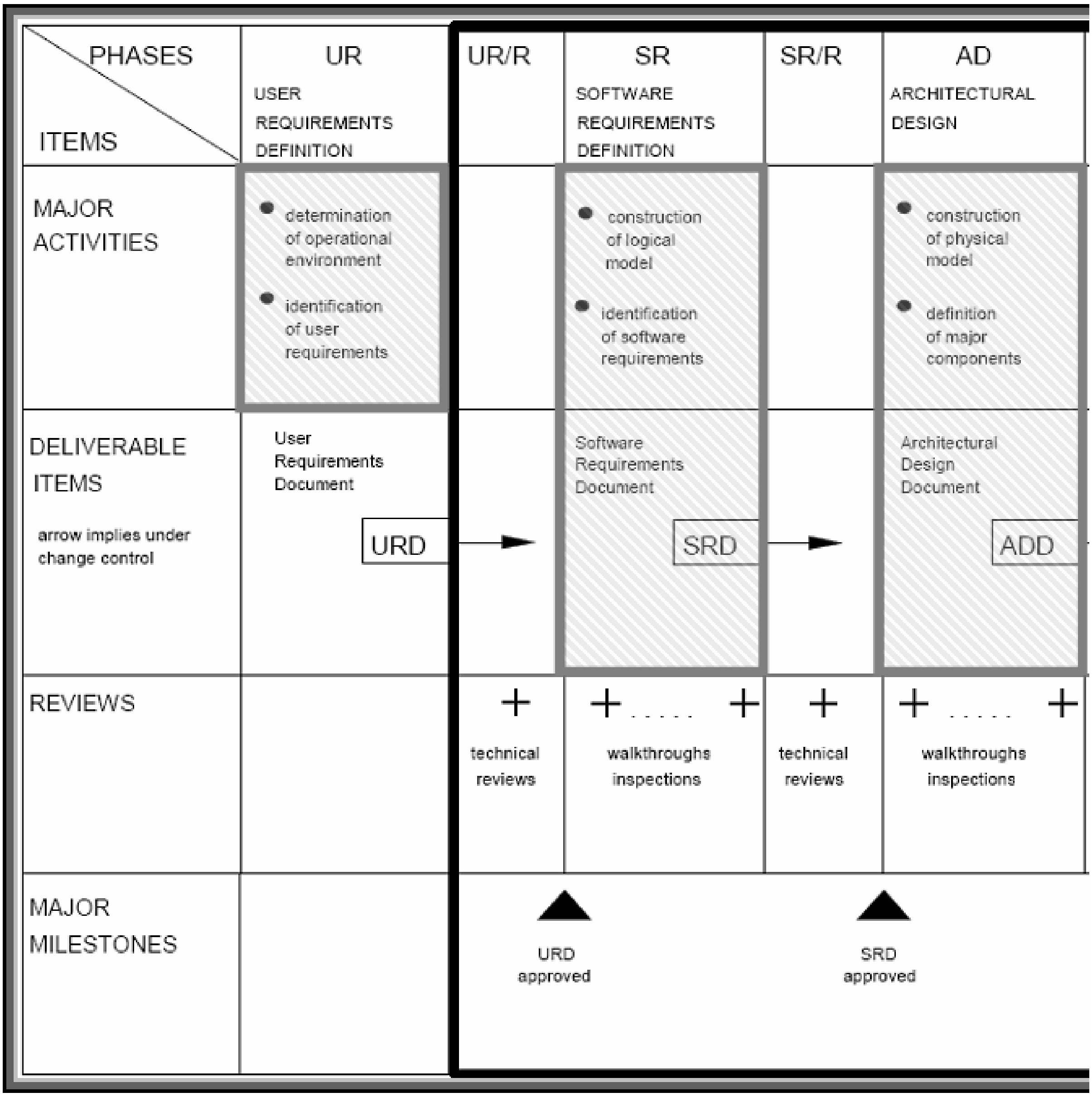
Using the methodology and the CPC reduces the UR phase of the software life cycle considerably. In fact, the only steps remaining are the review of the requirements, additional specifications and the writing of the URD.

To recapitulate, using the methodology and respectively the CPC is a good investment, since it has many benefits, and does not generate exhaustive costs. Actually, approximately an half of the meetings could be avoided by using the methodology. Only the meetings to define specific requirements remain. It has repercussion on the following aspects: accommodation costs, loss of time, cost generated through the presence of specialists and team members, creation of surveys and interviews, realisation of interviews, etc. Let us now consider the costs of applying the methodology. The personal costs of the one defining the project profile only must be taken into consideration which corresponds to 15 to 30 minutes of the project leader's time.

As become apparent, due to the low costs of using the methodology and the CPC, it presents advantages in almost every case. The initial investment of using it is so marginal, that it can be used, even if the recommendations provided do not correspond exactly to the project's needs.

5.2.4. Conclusion

As we have seen in 5.2.1 and 5.2.3, using a configured standard tool as a platform together with the methodology and the CPC reduces the time to application of the collaboration platform within an engineering project. Hence, you have the possibility to achieve drastic time savings. Major tasks, such as the construction of logical models and prototypes, architectural models, module design, coding, and unit tests, are not needed in part, if not completely, since they have already been carried out when developing the standard tool. Figure 5-3 and Figure 5-4 shows with respect to the software life cycle model those activities that can partially or completely fail. They are marked in grey.



PHASES / ITEMS	UR USER REQUIREMENTS DEFINITION	UR/R	SR SOFTWARE REQUIREMENTS DEFINITION	SR/R	AD ARCHITECTURAL DESIGN
MAJOR ACTIVITIES	● determination of operational environment ● identification of user requirements		● construction of logical model ● identification of software requirements		● construction of physical model ● definition of major components
DELIVERABLE ITEMS arrow implies under change control	User Requirements Document URD	→	Software Requirements Document SRD	→	Architectural Design Document ADD
REVIEWS		+ technical reviews	+ . . . . . + walkthroughs inspections	+ technical reviews	+ . . . . . + walkthroughs inspections
MAJOR MILESTONES		▲ URD approved		▲ SRD approved	

Figure 5-3: Extended software life cycle model, part 1(according to [ESA, 1991]).

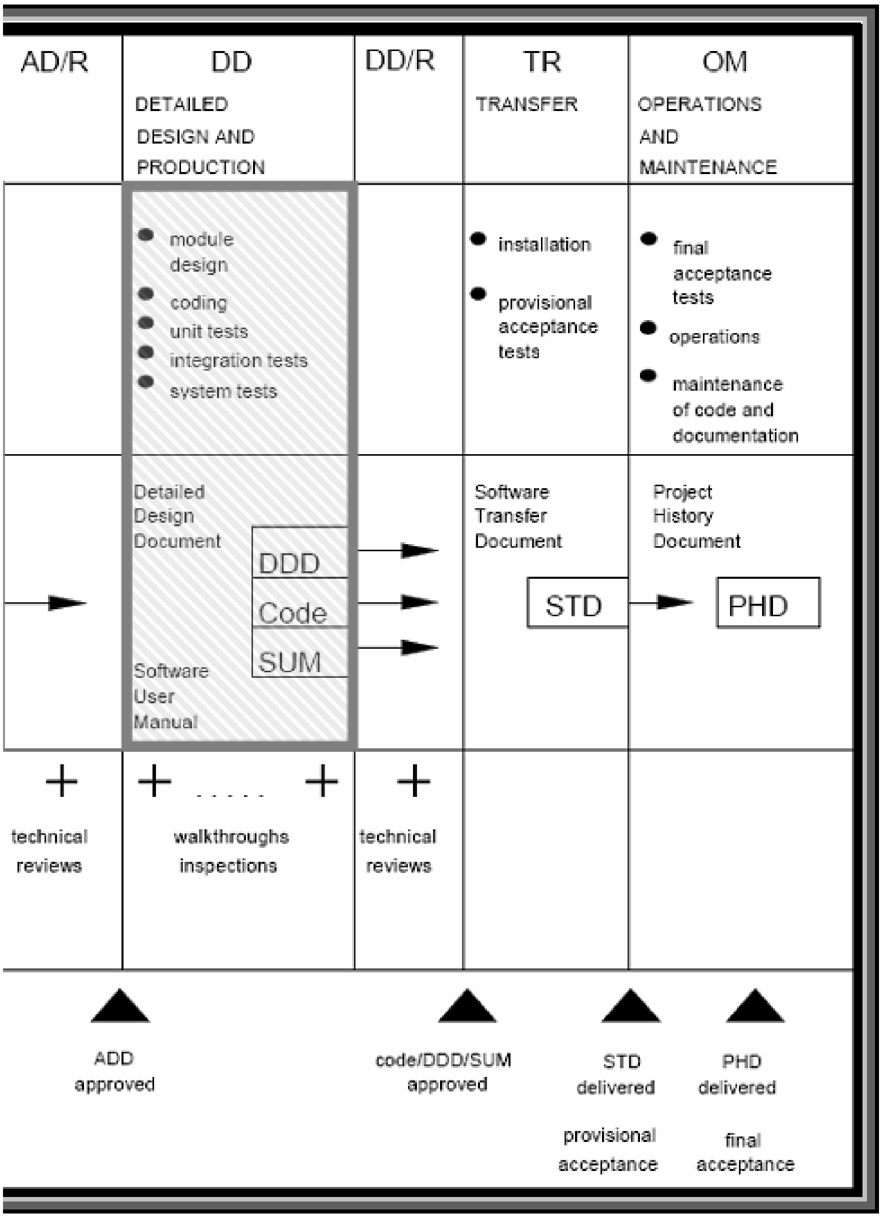


Figure 5-4: Extended software life cycle model, part 2(according to [ESA, 1991]).

Only a few activities remain: the user-requirement definition, which is simplified and cheaper through the use of the method, as well as the phases from the transfer phase. The purpose of this phase is to guarantee that the platform fulfills the requirements laid down in the URD. This is accomplished by installing and configuring it. Thereafter, acceptance tests are carried out. When the platform provides the required capabilities, it can be provisionally accepted and used within the project.

In this case, time to application is reduced of a minimum of 50%. The case study presented in 4.7.2 particularly highlights this aspect of the evaluation. In fact, according to [Hummel et al., 2005], in this software development project, it took almost one and an half years to get an optimal IT-support. With the help of the methodology and the standard tool, we configured an appropriate IT-collaboration-platform in less than one hour.

6. SUMMARY AND OUTLOOK

In this chapter we first summarize the work and results presented in the thesis. Secondly, we propose some ideas for optimising the developed methodology and expert-system as well as ideas for pursuing further research.

6.1. Summary

First, this thesis presents current trends in engineering: cross-enterprise and cross-domain engineering. After defining these terms and pointing out their principal characteristics and difficulties, Chapter 2 proposes a taxonomy of influencing factors of cross-enterprise and cross-domain projects.

Some of the challenges of cross-enterprise and cross-domain contexts can be solved by or with the help of sophisticated IT-tools and platforms. The main focus of IT-systems for CDE and CDD projects centers on topics like communication, the transfer of information, team coordination as well as special cases of the engineering process (cf. [Lassenius et al., 2003]) all of which presented in Chapter 3.

Although there is an extensive body of literature on CSCW, there is a lack of systematic knowledge about the influence of certain characteristics of distributed projects on project success and the strengths of these influences. The approach described in the thesis provides precisely such a set of project criteria and a stringent method to derive the optimal tool chain for a certain project based on the study of the influencing factors of CEE and CDE.

The method developed is based on fuzzy logic and provides a recommendation for configuring IT-collaboration-platforms specific to given project profiles. It follows the normal construction processes of fuzzy-expert-systems, i.e. it consists of an interface of fuzzification, the decision logic based on expert-knowledge, and an interface of defuzzification.

The interface of fuzzification transforms the characteristics of the project evaluated by the project leader into fuzzy values. Toward this end, the influencing factors have been represented through fuzzy sets, with the exception of the exact factors which are represented by exact values. Each factor definition has been justified separately, when possible with the aid of previous analyses, generally statistical ones and works relating about fuzzy applications. For the factors ‘Distribution model’, ‘Number of interfaces’, ‘Vocabulary’, ‘Methods and instruments’, and ‘Standards and laws’, we had to carry out an empirical study to define the fuzzy sets, since no previous reports or studies were available the creation of the sets could be based on.

The decision logic determines the tools appropriate for the given project configuration on the basis of an empirical study. In this study 18 standard IT-tools for supporting global development projects have been rated by experts in support of cooperation projects with respect to the defined influencing factors. The survey was carried out by interviewing experts in distributed projects, interdisciplinary projects, cooperation projects, and Information Technology (IT). To weigh the factors we executed a sensitivity analysis on the expert data which demonstrated that not always the same factors have an impact on the choice of one tool or another each tool. Each tool has its own characteristics and its implementation does not depend on the same boundary conditions of global development projects.

The interface of defuzzification converts the fuzzy output set to a result that can be understood by everyone. It returns a ranking list of the tools sorted by the fields 'very adapted', 'recommended', 'neutral', 'not recommended' and 'not adapted'. The method has been used to develop a small software application implemented at the BMW Group for planning the IT-support of cooperation projects.

The methodology has been applied to real product-development and software-engineering projects and the results implemented in VPS, a standard IT-platform for collaborative engineering integrated in the Partner Portal of the BMW Group. The selected projects displayed different aspects of the methodology which highlights the benefits of using it.

First, the resulting collaboration-platforms are optimised to the projects' needs and follow the current business and engineering processes. Second, using a configured standard tool as a platform as well as the developed methodology to configure it is of great significance to reduce "time to decision" processes in the early phase of a development project, especially if the project must start quickly and the managers do not have time to wait for an analysis of the project's configuration and possible IT-support. Since it reduces the development and implementation time of the platform within an engineering project, taking the methodology and the standard IT-platform result in drastic time-saving and a quicker estimate of the global IT-costs.

6.2. Outlook

In this section we propose different means of pursuing this research. First, how the methodology and the CPC could be optimised and, secondly, how the research on the configuration of user interfaces with fuzzy-expert-systems could be continued.

6.2.1. Improvements of the Methodology

The developed methodology presents a recommendation for configuring IT-collaboration-platforms for specific CEE and CDE projects. However, since the tools suggested are standard ones, they do not always correspond to the projects' specific needs. In fact, more often than not standard tools are not sufficient to support every project.

Subsequently, the methodology could, on the one hand, be improved with respect to the tools. It could first consider discipline specific tools as well as standard ones. Secondly, it is worth noting that another improvement of the methodology would be the consideration of every function's feature and not only its global functionality. Features of an electronic calendar would be, for example, meeting planner, resource planner, automatic notification, calendar sharing, connection of personal calendar to group calendar, etc.

The tool's features used in a project, especially if they are used by all participants or imposed by the project leader, have important repercussions on the processes that follow, e.g., when the project leader decides to use a document management system. Features of a document management system are: check-in/out functions, versioning, achieving, document workflow, document authorisation, document alert, filing of metadata, filtering, search, deposit of a structure, etc. The selection of, for example, the "alert" feature ensures that the persons concerned with a document receive notification when it has been disposed in the system. Since everyone is committed to take note of certain appointed documents, it influences, of course, the process followed in the project.

Another optimisation of the methodology and the CPC would be the creation of relations between the tools. To date, the expert-system correlates the influencing factors describing the project to IT-tools in order to give recommendations about which IT-tool is appropriate or not, but it does not make any correlation between the tools. Let us explain what we understand by correlation between tools, e.g.:

- If tool A is appropriate to your project, then tool E should not be integrated into your platform, since it satisfies nearly the same needs or wishes and would therefore be redundant.
- If tool A is appropriate to your project, then you should not use tool B, since they support incompatible processes.
- If tool A is appropriate to your project, then you might also want to use tools C and D, since they are complementary.

Thus far all the proposed improvements concern the IT-tools and require a more precise and time-consuming analysis of each tool and its features. However, it is important to note that IT changes at an ever-increasing rate (cf. [Oberlin, 1994]) and therefore casts these optimisations of the expert-system regarding features of tools into doubt.

Secondly, the methodology and CPC could be optimised by adding factors specific to an organisation or to the kind of products developed within it. Although it is not the object of this thesis to define a methodology for merely one organisation, it would nevertheless make sense, especially for large-scale enterprises, since they often have their own criteria of evaluation.

Another possibility to optimise the methodology is to enable several users to fill in the check-list and give the project configuration and then consolidate their results. This would reduce the risk of a distorted evaluation of a single person and consequently avoid a false decision.

To improve the recommendations made by the system we could also provide the user with the possibility of giving a feedback of the results and thus enable the system to correct itself, similar to a so-called “recommender system”.

On the whole, in a typical recommender system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients. In some cases the primary transformation is in the aggregation; in other words the system's value lies in its ability to make good matches between the recommenders and those seeking recommendations (cf. [Resnik et al., 1997]).

Our system could enable, for example, the user to tell us, whether he would have put the tool in a higher or lower position of ranking, or if he would like to use one of the tools not listed in the ranking. Thus the system would have the possibility to correct itself little by little and update the knowledge base. It would also provide more efficient results and enable a constant improvement of the system’s knowledge, which is of importance, since the attitude toward some tools may vary over the years with new emerging technologies.

The system could also give some advice on how the recommended tools have to be implemented, how the project’s participants have to be prepared for the changes occurring with the implementation, and make them notice points they should consider regarding their project profile. The Taxonomy System developed by Gierhardt gives this kind of advice, but does not relate it to the appropriate IT. As already mentioned, it points out typical problems of CEE and offers possible solutions for these problems (cf. [Gierhardt, 2001]).

6.2.2. Further Possible Research

Various possibilities exist for pursuing this research. As already mentioned in the last section, the study can be used for every domain. Instead of considering standard tools for collaboration, the user may consider every existing tool for his discipline, e.g. for software engineering he could have tools for configuration management, data modelling, problem tracking, document generation, reverse engineering/maintenance, testing, object oriented simulation, batch code analysis, audit, transformation, requirements based testing, requirements engineering, requirements tracing, GUI development, collaborative Internet-based requirements management, database applications, etc. The lists of CASE tools (Computer-Aided Software Engineering) is exhaustive. [Lamb, 2003] alone listed more than 600 existing CASE tools. Subsequently, it makes sense to develop a methodology that gives recommendations about when you should use a specific tool or not.

On a more abstract level, the same concept can be used to configure not only IT-platforms for collaboration, but also, for example, user interfaces in general. Of course, toward this end the factors must be redefined depending on the underlying product of the user

interface. Subsequently, the repertoire of functionalities must be newly created in the empirical knowledge-base.
Let us take an example. Currently, cell phones offer vast possibilities for obtaining and using information mobile, although they also present major design challenges. According to [Golvin et al., 2006], consumers faced with so many technology choices are forced to ask themselves when a device crosses the line from manageable multitasker to operational overload. The concept developed in this thesis could be used to develop a cell phone configuration-tool. Depending on influencing factors, such as the age of the user, his habits, his experience with cell phones, the common gadgets he normally uses, if he has a computer, an MP3-Player, the expert-system, determines a set of functionality appropriate for the user and his environment. Such a system could be of use in a cell phone shop for considering the profile of the buyer and his personal needs immediately or at a cell phone maker to design new product lines of cell phones which correspond to different profiles of customers, which have been established by marketing departments.

Another way to pursue this research would be to apply the same method to other characteristics of IT. Instead of configuring the platform regarding the IT-tools' global functionality, the proposed research would take other aspects into consideration, such as architectural, security, safety, maintainability, quality aspects, etc.
We have seen in Chapter 5 that our methodology reduces the user requirements phase of the development of IT-collaboration-platforms in providing advices to configure it. In fact, the developed method aids the project leader in defining the functional user requirements of IT-collaboration-platforms. However, it would also be possible to consider other aspects of IT and thus to apply the method developed for IT-collaboration-platforms in the first place to other kind of requirements, such as quality or security requirements.

This way of pursuing research is not only limited to the development of collaboration-software, but can be extended to software engineering in general. Approaches of requirements engineering with fuzzy logic are very well imaginable. In fact, existing requirement methodologies are often limited in specifying requirements that are usually vague and imprecise (cf. [Yen et al., 1996]). That is why, for example, [Sora et al., 2006] proposes a fuzzy-logic based solution for the specification and retrieval of software components. Similarly, [Yen et al., 1996] has developed a fuzzy-logic based systematic tradeoff methodology for acquiring and validating imprecise requirements.

The methodology to be developed could, for example, help producing a set of software requirements that is complete, consistent and as correct as possible regarding the different types of requirements: functional requirements, performance requirements, interface requirements, operational requirements, resource requirements, verification requirements, acceptance testing requirements, documentation requirements, security requirements, portability requirements, quality requirements, reliability requirements, maintainability requirements, and safety requirements.

E.g. let us consider quality requirements. In [Deissenboeck et al., 2007] the authors state that quality can not be considered just for its own good, but that quality requirements have to be discussed and prioritised within an economic context. Therefore, it makes sense to derive the quality requirements directly from the business goals at the beginning of the project (cf. [Geisberger et al., 2006]). In this context we could imagine a methodology similar to the one developed in this thesis which helps in the prioritisation of quality requirements regarding the business goals. Quality requirements are, for example, maintainability, usability, flexibility, correctness, etc. If the most important business goals are, for example, efficiency, customer satisfaction, and safety goals, we can then easily derive that usability requirements, such as simplicity, customizability, adaptability, etc., will be significant in the requirements definition (cf. [Winter et al., 2007]). In contrast, if an important business goal consists of the suitability of the software for reuse in a different context, then reusability or maintainability requirements will be more significant.

The method to be developed will then use a knowledge-base which correlates business goals with quality requirements. This knowledge-base should be completed by experts in quality engineering in the domain of the software. In practice, the stakeholders of the software to be developed will fill in a checklist asking them to which degree they want to attain a specific goal and get then the prioritisation of the requirements as a response. Therefore, a fuzzy-expert-system would be very appropriate, since they possibly want to attain a goal only to some extent.

7. LITERATURE

[AIDC, 2006] AIDC, Mechatronics: The New Language of the Automobile, *Motor Age, 2006*

[Allen, 1984] Allen, T. J., Managing the Flow of Technology, *Cambridge (MA): MIT Press 1984*

[AMT, 2007] Akademie für Management und Technik GmbH, *http://www.amt-wiesbaden.de/index.php?ziel=1amt14#2, 2007*

[Anderl et al., 1999a] Anderl, R.; Lindemann, U.; Thomson, B.; GAUL, H.-D.; Gierhardt, H.; Ott, T., Investigation of Distributed Product Design and Development Processes, *Proceedings of the 12th International, Conference on Engineering, 1999*

[Anderl et al., 1999b] Anderl, R.; Ott, T.; Lindemann, U.; Gaul, H.-D.; Gierhardt, H., Planning and Improvement of Distributed Product Development Processes by Using a Taxonomy System, *Concurrent Engineering: From Product Design to Product Marketing. 6th European Concurrent Engineering Conference, 1999*

[Arisha et al., 1999] Arisha, K.A., Ozcan, F., Ross, R., and Subrahmanian V.S.: Impact: A Platform for Collaborationg Agents, *IEEE Intelligent Systems, 1999*

[Astley et al., 1998] Astley M., Agha, G.A, Customization and Composition of Distributed Objects: Middleware Abstractions for Policy Management, *ACM SIXSOFT 6th International Symposium on Foundations for Software Engineering, 1998*

[Bannon, 1992] Bannon, L., Interdisciplinarity or Interdisciplinary Theory in CSCW?, *CSCW´92 Workshop on Interdisciplinary Theory for CSCW Design, Toronto, Canada, 1992*

[Baumann et al., 2001] Baumann, M., Heinen, E., Holzbach, W.:, Innovative Dienstleistungen im Handwerk, Konzeptentwicklung und Praxisbeispiele, *Ergebnisse eines vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Projektes. Gifhorn, 2001*

[Bordeau et al., 1997] Bordeau, J., Wasson, B., Orchestrating collaboration in collaborative telelearning*, in: Artificial intelligence in education, IOS Press, 1997*

[Bornschein et al., 1995] Bornschein-Grass, C., Picot, A., Reichwald, R., Groupware und computerunterstütze Zusammenarbeit - Wirkungsbereiche und Potentiale, *Wiesbaden, Gabler Edition Wissenschaft, 1995*

[Brehmer, 1991] Brehmer, B., Distributed decision making: some notes on the literature, *Chichester, England ; New York, Wiley, 1991*

[Brown, 2006] Brown, J., Product Lifecycle Collaboration Benchmark Report: The Product Profitability “X Factor”?, *Aberdeen Group, 2006*

[Buchanan et al., 1988] Buchanan, B.G., Smith, R.G., Fundamentals of Expert-systems, *Annual Review of Computer Science, Vol. 3, 1988*

[Carstensen et al., 2002] Carstensen, P. H., Schmidt, K., Computer supported cooperative work: New challenges to systems design*, Handbook of Human Factors. K. Itoh. Tokyo, 2002*

[Chang et al., 2001] Chang, C., Zhang, J. Jiang, T.M., Formalization of Computer Supported Cooperative Work Applications, *8th IEEE Workshops, 2001*

[Code Zone, 2005] Code Zone 00001001, Google Translator Web Part, *http://www.00001001.ch/Resources/Code/1860.aspx, 2005*

[Cortes et al., 1996] Cortes, M., Mishra, P.: DCWPL: A programming Language For Describing Collaborative Work, *Proceedings of the ACM 1996 Conference on CSCW, Cambridge MA USA, 1996*

[Crystal, 1997] Crystal, D., Watching World English grow, *IATEFL Newsletter; adapted for Concorde 5, English Speaking Union, 1997*

[Danner, 1996] Danner, S., Ganzheitliches Anforderungsmanagement mit QFD – ein Beitrag zur Optimierung marktorientierter Entwicklungsprozesse, *Aachen: Shaker 1996, (Konstruktionstechnik München, Band 24), München: TU, Diss. 1996*

[DB, 2007] Deutsche Bahn AG, *http://www.bahn.de, 2007*

[Deissenboeck et al., 2007] Deissenboeck, F., Wagner, S., Kosten-basierte Klassifikation von Qualitätsanforderungen, Erhebung, Spezifikation und Analyse nichtfunktionaler Anforderungen in der Systementwicklung, *SE Konferenz, Hamburg, 2007*

[Deresky, 2000] Deresky, H., International Management, Managing across Borders and Cultures, *Prentice Hall, Upper Saddle River, New Jersey, 2000*

[Dillenbourg et al., 1995] Dillenbourg, P., Baker, M., Blaye, A., O'Malley, C., The Evolution of Research on Collaborative Learning. Learning in humans and machines*, in: Towards an interdisciplinary learning science, London, Pergamon, 1995*

[DIN, 2000] DIN German Institute for Standardization e. V., Economic benefits of standardization, *Beuth Verlag, 2000*

[Dubois et al., 1978] Dubois, D., Prade, H., 'Operations on fuzzy numbers', *International Journal of Systems Science, 1978*

[Dubois et al., 1979] Dubois, D., Prade, H., 'Fuzzy real algebra: some results', *Fuzzy Sets and Systems, 1979*

[Dubois et al., 1996] Dubois, D., Prade, H., Using fuzzy sets in flexible querying: Why and how?, *Workshop on Flexible Query-Answering Systems, Denmark, 1996*

[Dubois et al., 2002] Dubois, D., Prade, H., La problématique scientifique du traitement de l´information, *Revue 13, Vol. 1, N°2, 2002*

[EC-GI&GIS, 2001] Ford, M., Evmorfopoulou, C., 7th EC-GI&GIS Workshop, *Potsdam, 2001*

[Ehrlenspiel, 1995] Ehrlenspiel, K., Integrierte Produktentwicklung, Methoden zur Prozessintegration, Produkterstellung und Konstruktion, *Carl Hanser, München, 1995*

[Eigner et al., 2006] Eigner, M., Schleidt, B., Der ‘Faktor Mensch’ im Cross-Enterprise Engineering, *ProSTEP iViP Symposium, 2006*

[Eigner et al., 2007] Eigner, M., Ovtcharova, J., Produktentstehung im 21. Jahrhundert, *Digital Engineering Magazin, 2007*

[Einstein et al., 1921] Einstein, A., Janssen, M., Schulmann, R., et al., eds., The Collected Papers of Albert Einstein, The Berlin Years: Writings, 1918-1921, Geometry and Experience, *Berlin, 1921*

[Ellis et al., 1991] Ellis, C. A., Gibbs, S. J., Rein, G., Groupware: Some issues and experiences, *Communications of the ACM, 34(1), 1991*

[Encarnação, 2005] [Encarnação, J.L., Editorial: Cross-Domain Engineering, *ProduktDatenJournal, ProSTEP iVIP, 2005*

[Engelmore et al., 1993] Engelmore, R.S.,Feigenbaum, E., Friedland P.E., Johnson, B.B., Nii, H.P., Schorr, H., Shrobe, H., Knowledge-Based Systems in Japan, *Japanese Technology Evaluation Center, 1993*

[ENSR, 2003] ENSR - European Network for SME Research, Les PME en Europe en 2003, *Observatoire des PME européennes, No. 7, 2003*

[ESA, 1991] ESA, Software engineering standards Issue 2, *European Space Agency / Agence Spatiale Européenne, Paris, 1991*

[Fruchter et al., 1995] Fruchter, R., Leifer, L.J., Toye, G., Reiner, K.A., Collaborative Mechatronic System Design, *CERA, 1995*

[Garbay, 2006] Garbay C., Les sciences du traitement de l´information comme pivot de l´interdisciplinarité : une vision systématique, *CNRS, 2006*

[Gaul, 2001] Gaul, H.D., Verteilte Produktentwicklung – Perspektiven und Modell zur Optimierung, *TU München Diss. 2001*

[Geisberger et al., 2006] Geisberger, E., Broy, M., Berenbach, B., Kazmeier, J., Paulish, D., Rudorfer, A., Requirements Engineering Reference Model (REM), *Technischer Bericht, Technische Universität München, 2006*

[Gierhardt et al. 1999] Gierhardt, H.; Gaul, H.-D.; Ott, T., Distribution in Product Design and Development Processes, *Proceedings of the 1999 ASME Design Engineering Technical Conferences and Computers in Engineering Conference, 1999*

[Gierhardt, 2001] Gierhardt, H., Global verteilte Produktentwicklungsprojekte – Ein Vorgehensmodell auf der operativen Ebene, *München: Dr. Hut 2001. (Produktentwicklung München, Band 46), TU München Diss. 2001*

[Golvin et al., 2006] Golvin, C., Schadler, T., Portable Multifunction Devices: What Works? Only Activity-Focused Multifunction Devices Will Succeed*, Forrester Research*, *2006*

[Google, 2007] Google Language Tools, *http://www.google.com/language_tools?hl=en, 2007*

[Greenberg, 1991] Greenberg, S., Computer-supported Cooperative Work and Groupware*, London, Academic Press Ltd, 1991*

[Griffin, 1999] Griffin, R. W., Pustay, M. W., International Business, a Managerial Perspective, *Addison-Wesley, Reading , Massachusetts, 1999*

[Grudin, 1994] Grudin, J., Computer-supported cooperative work: Its history and participation.' *IEEE Computer 27(5), 1994*

[Gulp, 2007a] GULP - Das Portal für IT-Projekte, GULP Stundensatz Kalkulator, *http://www.gulp.de/kb/tools/money.html, 2007*

[Gulp, 2007b] GULP - Das Portal für IT-Projekte, Stundensatz Teil 7, *http://www.gulp.de/kb/st/stdsaetze/umfrageVII_f.html, 2007*

[Harms, 2005] Harms, H.H., Simultaneous Engineering bei der Entwicklung von modernen Landmaschinen, *Business Talks, Hannover Messe, 2005*

[Hartmann, 1998] Hartmann, Y., Controlling Interdisziplinärer Forschungsprojekte, *1998*

[Hasenkamp et al., 1994] Hasenkamp, U.; Syring, M., CSCW (Computer Supported Cooperative Work) in Organisationen - Grundlagen und Probleme; *in CSCW - Computer Supported Cooperative Work - Informationssysteme für dezentralisierte Unternehmensstrukturen, Bonn, Addison-Wesley Verlag, 1994*

[Heckhausen, 1987] Heckhausen, H., Interdisziplinäre Forschung: zwischen Intra-, Multi- und Chimären-Disziplinarität, *Praxis - Herausforderung – Ideologie, Frankfurt a.M., 1987*

[Hohl, 2005] Hohl, B., Fuzzy-Summierbarkeit, *Universität Karlsruhe, 2005*

[Hoult, 1969] Hoult, T. F., Dictionary of Modern Sociology, *Totowa, New Jersey, United States: Littlefield, Adams & Co., 1969*

[Hübenthal, 1991] Hübenthal, U., Interdisziplinäres Denken: Versuch einer Bestandsaufnahme und Systematisierung, *Stuttgart, 1991*

[Hummel et al., 2005] Hummel, B., Stangl, H., Global Studio Project - First Phase, *Technische Universität München, 2005*

[IGD, 2004] Fraunhofer-Institut für Graphische Datenverarbeitung (IGD), *http://www.igd.fhg.de, Darmstadt, 2004*

[Iglsböck, 2002] Iglsböck, E., Science News: Synergy benefits from symbiosis, *BMW Science Club, München, 2002*

[Inverstor, 2006] InvestorWords, The biggest, best investing glossary on the web, *http://www.investorwords.com/, 2006*

[ISO 9241, 1998] International Organization for Standardization, ISO 9241, Part 11: Guidance on Usability, *1998*

[Jantzen, 1998] Jantzen J., Tutorial on Fuzzy Logic, *Tech. report no 98-E 868, Technical University of Denmark, 1998*

[Jennex, 2005] Jennex, M., Knowledge Management Systems, *San diego University, 2005*

[Jeppson et al., 1999] Jeppsson, P., Johanson, M., Stenius, M., Törlind, P., Collaboration Environments for Distributed Engineering Development of a Prototype System, *presented on Computer Supported Cooperative Work in Design 99, France, 1999*

[Johannsson, 2004] Johannsson, F., The Medici Effect – Breakthrough Insights at the Intersection of Ideas, Concepts & Cultures, *Harvard Business School Press, 2004*

[Johansen, 1988] Johansen, R., Groupware - computer support for business teams*, New York, Free Press, 1988*

[Kaufmann et al., 1991] Kaufmann, A., Gupta, M., Introduction to Fuzzy Arithmetic, Theory and Applications, *Van Nostrand Reinhold, New York, 1991*

[Kaufmann, 1975] Kaufmann, A., Introduction to the Theory of Fuzzy Subsets, vol. 1, *1975*

[Kazi et al., 2001] Kazi, A.S., Hannus, M., Laitinen, J., Nummelin, O., Distributed Engineering in Construction: Findings from the IMS Globemen Project, *ITcon Vol.6, 2001*

[Keil et al., 2006] Keil, P., Milewski, A. E., Mullick, N., Richardson, I., Distributed Development - an Education Perspective on the Global Studio Project, *Proc. of the 28th International Conference on Software Engineering (ICSE), Shanghai, China, 2006*

[Kenney et al., 1962] Kenney, J. F., Keeping, E. S., "The Standard Deviation" and "Calculation of the Standard Deviation.", *Mathematics of Statistics, Pt. 1, 3rd ed. Princeton, 1962*

[Kirwin, 1995] Kirwin, C., Reasoning, *In Ted Honderich (ed.): The Oxford Companion to Philosophy, Oxford University Press, 1995*

[Klir et al., 1997] Klir, G., UTE H., Bo Y., Fuzzy Set Theory Foundations and Applications,*1997*

[Kreimeyer et al., 2006] Kreimeyer, M., Lindemann, U., Deubzer, F., Herfeld, U., Maier, A., Clarkson, P.J., Reflecting Communication: A Key Factor for Successful Collaboration between Embodiment Design and Simulation, *Marjanovic, D.: 9th International Design Conference Dubrovnik, 2006*

[Kruse et al., 1995] Kruse, R., Gebhardt, J., Klawonn, F., Fuzzy-Systeme (2. Auflage), *Teubner, Stuttgart, 1995*

[Lakatos, 1970] Lakatos, I., Falsification and the Methodology of Scientific Programmes, *Cambridge University Press, Cambridge, U.K., 1970*

[Lamb, 2003] Lamb, D.A., Software Engineering Archives, http://www.cs.queensu.ca/Software-Engineering, *Queen's University, Canada, 2003*

[Lassenius et al., 2003] Lassenius, C., Paasivaara, M., Collaboration practices in global inter-organizational software development projects, *Software Process Improvement and Practice 8(4), 2003*

[Laurent, 2006] Laurent, C., Design of IT-Collaboration-Platforms with Fuzzy Logic, *Current Research in Information Sciences and Technologies Multidisciplinary approaches to global information systems, Open Institute of Knowledge, ISBN: 84-611-3106-1, 2006*

[Laurent, 2007a] Laurent, C., Analysis of the Survey: 'IT-Platforms for Cooperation Projects', *Technical Report, TU München, 2007*

[Laurent, 2007b] Laurent, C., A Sensitivity Analysis Approach to Select IT-Tools for Global Development Projects, *International Conference on Global Software Engineering, REMIDI Workshop, 2007*

[Li et al., 1998] Li, D., Muntz, R., COCA: Collaborative Objects Coordination Architecture, Work, *Proceedings of the ACM Conference on CSCW, USA, 1998*

[Lo et al., 2006] Lo, C.C., Wanga, P., Chao, K.-M., A fuzzy group-preferences analysis method for new-product development, *Expert-systems with Applications, Volume 31, Issue 4 , 2006*

[McComb et al., 1999] McComb, S., Green, S., Compton, W., Project goals, team performance, and shared understanding, *Engineering Management Journal, 11(3), 1999*

[McNamara, 1999] McNamara, C, Basic Definition of Organization, *Pad, 1999*

[Mendel, 2001] Mendel, J., Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, *New Jersey, 2001*

[Mertens, 1995] Mertens, P., Faisst, W., Virtuelle Unternehmen - eine Organisationsstruktur für die Zukunft?, *technologie & management 44, 1995*

[Microsoft, 2004] Microsoft Corporation, Integrating Exchange Server 2003 with SharePoint Products and Technologies, *http://www.microsoft.com/technet/windowsserver/sharepoint/v2/reskit/c4161881x.mspx, 2004*

[Milberg et al., 1996] Milberg, J.; Reinhardt, G.; (Hsrg.): Time to Market, Seminarberichte 16, *Institut für Werkzeugmaschinen und Betriebswissenschaften, Technische Universität München, 1996*

[Mills, 1998] Mills A., Collaborative Engineering and the Internet, *1998*

[Mizumoto, 1989] Mizumoto, M., Improvement methods of fuzzy controls, *Proc. of 3rd IFSA Congress (August 6-11, 1989, Seattle), 1989*

[Morris, 1991] Morris, M.D., Factorial Sampling Plans for Preliminary Computational Experiments, *Technometrics, 1991*

[NRC, 2006] NRC Institute for Information Technology, FuzzyJ ToolKit for the Java(tm) Platform & FuzzyJess, *http://www.iit.nrc.ca/IR_public/fuzzy/fuzzyJToolkit2.html, 2006*

[Oberlin, 1994] Oberlin, J.L., Departmental Budgeting for Information Technology: A Life-cycle Approach, *CAUSE From CAUSE/EFFECT, Volume 17, Number 2, 1994*

[Okujava, 2006] Okujava, S., Wirtschaftlichkeitsanalysen für IT-Investitionen - Ein kontinuierlicher und stakeholderorientierter Ansatz, *WiKu Verlag, 2006*

[Olss, 2000] Online Survey Services, Cost Comparison of Paper Survey vs. Online Survey, *http://www.onlinesurveyservices.com/olss/cost.htm, 2000*

[Pau, 2002] Pau, L.-F., IPv6 Return on investment (R.O.I) analysis framework at a generic level, and first conclusions, *Erasmus Research Institute of Management (ERIM), 2002*

[Persson et al., 2005]	Persson, A., Lings, B., Lundell, B., Mattsson, A., Ärlig, U., Communication, coordination and control in distributed development: an OSS case study, *Proc. 1st International Conference on Open Source Systems, Genova, Italy, 2005*
[Petrie, 1998]	Petrie, C., Process Coordination, White Paper, *Stanford University, 1998*
[Picot et al., 1998]	Picot, A; Reichwald, R.; Wigand, R.-T., Die grenzenlose Unternehmung, Information, Organisation und Management, *Wiesbaden, 1998*
[PMI, 2004]	Project Management Institute, A guide to the project management body of knowledge (PMBOK guide), *Newtown Square, 2004*
[Probst et. Al., 1997]	Probst, G., Raub, S., Romhardt, K., Wissen Managen, Wie Unternehmen die wertvollste Ressource nutzen, *Gabler, Wiesbaden , 1997*
[Reichwald, 1999]	Reichwald, R., Moslein, K., Sachenbacher, H., Englberger, H., Oldenburg, S., Telekooperation. Verteilte Arbeits- und Organisationsformen, *Berlin, Springer 1998*
[Resnick et al., 1997]	Resnick, P., Hal, V. Recommender Systems, *Introduction to special section of Communications of the ACM, vol. 40(3), 1997*
[Richei, 1998]	Richei, A., Bewertung und Optimierung der Mensch-Maschine-Schnittstelle unter Nutzung der Fuzzy Set Theorie, *Bochum, 1998*
[Ritchey, 1998]	Ritchey, T., General Morphological Analysis, A general method for non-quantified modelling, *16th EURO Conference on Operational Analysis, 1998*
[Saltelli et al., 2000]	Saltelli, A., Chan, K., Scott, E.M., Sensitivity Analysis, *John Wiley & Sons publishers, Probability and Statistics series, 2000*
[Saltelli et al., 2004]	Saltelli A. Tarantola S., Campolongo, F., Ratto, M., Sensitivity Analysis in Practice. A Guide to Assessing Scientific Models, *John Wiley & Sons publishers, Probability and Statistics series, 2004*

[Saltelli, 2004]	Saltelli, A., Global Sensitivity Analysis: An Introduction, *Proc. 4th International Conference on Sensitivity Analysis of Model Output (SAMO '04), Los Alamos National Laboratory, 2004*
[Scherber, 1997]	Scherber S., Modellierung und Simulation mechatronischer Systeme, *Technischer Bericht Sie-97-02-001, 1997*
[Schmid, 2005]	Schmid, C., Course on Dynamics of multidisplicinary and controlled Systems, *Ruhr-Universität Bochum, 2005*
[Schrötter, 2002]	Schrötter, K., Wissen muss prozessorientiert organisiert werden, *Wissensmanagement online, 2002*
[Segebart et al., 2000]	Segebart, D., Schönenberg, R., Knowledge Grows if You Share it...Potentials and Problems of Interdisciplinary Research and Inter-institutional Cooperation, *German-Brazilian Workshop on Neotropical Ecosystems – Achievements and Prospects of Cooperative Research, Hamburg, 2000*
[SimLab, 2005]	SimLab 2.2., *http://webfarm.jrc.cec.eu.int/uasa/primer/index.asp, 2005*
[Smart-Research, 2007]	Smart-Research GmbH, Online Markt- und Meinungsforschung, *http://www.smart-research.de/Online-Panel.8.0.html, 2007*
[Smith et al., 1991]	Smith, P.G.; Reinertsen, D. G., Developing Products in Half the Time, *New York, 1991*
[Sora et al., 2006]	Sora, I., Todinca, D., Using Fuzzy Logic for the Specification and Retrieval of Software Components, *3rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence, Romania, 2006*
[Sperber, 2003]	Sperber, D, Why Rethink Interdisciplinarity?, *Rethinking Interdisciplinarity, Society of Information, C.N.R.S, 2003*
[Stenmark, 1999]	Stenmark, D., The duality of email as corporate information channel', *Internal Communication, Issue 49, October 1999*
[SurveyPro, 2006]	The Survey Professionals, Advanced Technology for Online Surveys, *http://www.surveypro.com/jsp/home/pricing.jsp?PageMode=OUT_Pricing, 2006*

[Tang et al., 1994] Tang, J.C., Isaacs, E., Rua, M., Supporting distributed groups with a montage of lightweight interactions, *Proc. ACM Conf. on Computer-Supported Cooperative Work, Chapel Hill, NC, 1994*

[Tchounikine, 2002] Tchounikine, P., Quelques éléments sur la conception et l´ingénierie des EIAH, *Actes des deuxièmes assises nationales du GdR, 13, Nancy, 2002*

[Teufel et al., 1995] Teufel, S., Sauter, C., Muhlherr, T., Bauknecht, K., Computerunterstützung für die Gruppenarbeit, *Addison-Wesley, Bonn, 1995*

[Treadwell, 1995] Treadwell, A., Fuzzy Set Theory Movement in the Social Sciences*, Public Administration Review, volume 55, issue 1, 1995*

[VDA, 2001] VDA, VDA-Empfehlung 4961/2 - Kooperationsmodelle und SE-Checkliste zur Abstimmung der Datenlogistik in SE-Projekten. *Forschungsbericht, Verband der Automobilindustrie e.V., 2001*

[Wagner, 2007] Wagner, S., Cost-Optimisation of Analytical Software Quality Assurance, *Technische Universität München, 2007*

[Webster, 2002] Webster's Third New International Dictionary, *Unabridged, Merriam-Webster, http://unabridged.merriam-webster.com, 2002*

[Weinberg, 1960] Weinberg, R. J., An Examination of Logical Positivism, *Harcourt, Brace and Company, New York, 1960*

[Wheelright et al., 1992] Wheelright, S.C.; Clark, K.B., Revolutionizing Product Development: Quantum Leaps in Speed Efficiency and Quality, *New York, 1992*

[Wildemann, 1999] Wildemann, H., Produktklinik: Wertgestaltung von Produkten und Prozessen*, München, 1999*

[Wilson, 1991] Wilson, P., Computer supported cooperative work : an introduction, *Oxford, England Norwell, MA, Intellect, Kluwer Academic Publishers, 1991*

[Winter et al., 2007] Winter, S., Wagner, S., Deissenboeck, F., A Comprehensive Model of Usability, *Proceedings of Engineering Interactive Systems, 2007*

[Yaskawa, 2006] Yaskawa Electric Corporation, *http://www.yaskawa.co.jp/en/, 2006*

[Yen et al., 1996] Yen, J., Tiao, W.A., Liu, X.F., A Systematic Tradeoff Methodology for Acquiring and Validating Imprecise Requirements, *Proceedings of the Fifth IEEE International Conference on, Vol.1, 1996*

[Zadeh, 1965] Zadeh, L.A., Fuzzy Sets, *Information and Control 8, 1965*

[Zhang et al., 2006] Zhang, S., Gua, N., Yanga, J., An norm-driven state machine model for CSCW systems, *Expert-systems with Applications, Volume 31, Issue 4 , 2006*

[Zimmermann, 1991] Zimmermann, H.J., Fuzzy set theory and its applications, *2nd ed., Kluwer Academic Publishers, Dordrecht, 1991*

[Zizala, 2006] Zizala, L., Ein IT-Package zur Anbindung externer Kooperationspartner, *TU München, BMW Group, 2006*

[Zwicky, 1969] Zwicky, F., Discovery, Invention, Research - Through the Morphological Approach, *The Macmillian Company, 1969*

8. APPENDICES

8.1. Definition of the Linguistic Variables (Collected Data)

	Distribution Model		Interfaces			
	equal	unequal	low	average	high	max
min	0	25	1	3	3	5
max	40,00	40,00	5,00	10,00	18,00	100,00
average value	**18,75**	**37,50**	**3,15**	**6,62**	**12,54**	**36,15**
sample variance	113,61	21,30	1,76	4,24	27,82	705,62
standard deviation	10,66	4,62	1,33	2,06	5,27	26,56
standard error	2,96	1,28	0,37	0,57	1,46	7,37
68%: -1σ	8,09	32,88	1,83	4,56	7,26	9,59
68%: 1σ	29,41	42,12	4,48	8,67	17,81	62,72
values outside 1σ	4	2	4	3	4	3
% values inside 1σ	69,23%	84,62%	69,23%	76,92%	69,23%	76,92%
95%: - 2σ	-2,57	28,27	0,50	2,50	1,99	-16,97
95%: 2σ	40,07	46,73	5,81	10,73	23,09	89,28
values outside 2σ	0	1	0	0	0	1
% values inside 2σ	100,00%	92,31%	100,00%	100,00%	100,00%	92,31%
99,7%: - 3σ	-13,23	23,65	-0,83	0,44	-3,29	-43,54
99,7%: 3σ	50,73	51,35	7,14	12,79	28,36	115,84
values outside 3σ	0	0	0	0	0	0
% values inside 3σ	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%

	Vocabulary			Methods and instruments		Standards	
	same	different	contradictory	same	different	same	different
min	30	20	0	50	30	80	30
max	90,00	80,00	80,00	95,00	75,00	100,00	99,00
average value	**73,08**	**48,85**	**22,15**	**82,69**	**56,15**	**90,38**	**69,15**
sample variance	244,38	362,13	668,75	133,14	146,79	36,39	371,67
standard deviation	15,63	19,03	25,86	11,54	12,12	6,03	19,28
standard error	4,34	5,28	7,17	3,20	3,36	1,67	5,35
68%: -1σ	57,44	29,82	-3,71	71,15	44,04	84,35	49,88
68%: 1σ	88,71	67,88	48,01	94,23	68,27	96,42	88,43
values outside 1σ	3	4	3	3	4	3	4
% values inside 1σ	76,92%	69,23%	76,92%	76,92%	69,23%	76,92%	69,23%
95%: - 2σ	41,81	10,79	-29,57	59,62	31,92	78,32	30,60
95%: 2σ	104,34	86,91	73,87	105,77	80,39	102,45	107,71
values outside 2σ	1	0	1	1	1	0	1
% values inside 2σ	92,31%	100,00%	92,31%	92,31%	92,31%	100,00%	92,31%
99,7%: - 3σ	26,18	-8,24	-55,43	48,08	19,81	72,29	11,32
99,7%: 3σ	119,97	105,94	99,73	117,31	92,50	108,48	126,99
values outside 3σ	0	0	0	0	0	0	0
% values inside 3σ	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%

8.2. Results of the sensitivity analysis

We have realised the sensitivity analysis with the tool SimLab, the setting ‘Factor Prioritisation’, the method FAST and 50000 random samples (see 4.4.2).

	Workflow system		**Whiteboard**	
	Index	**%**	**Index**	**%**
Number of cooperation partners	0,0836	9,39%	0,0175	1,99%
Location	0,0189	2,12%	0,2941	33,37%
Skill level in the agreed language	0,0073	0,82%	0,0179	2,03%
Type of engineering process	0,00000241	0,00%	0,0243	2,76%
Organisation and company's culture	0,0019	0,21%	0,0142	1,61%
Size of the organisation	0,0734	8,24%	0,0068	0,77%
Intensity of collaboration	0,0582	6,53%	0,1579	17,91%
Distribution model	0,0161	1,81%	0,0653	7,41%
Number of interfaces	0,0566	6,36%	0,0029	0,33%
Access to data	0,0215	2,41%	0,0202	2,29%
Skill level in lthe use of IT	0,3003	33,72%	0,0542	6,15%
Influence of time	0,0475	5,33%	0,1467	16,64%
Vocabulary	0,0896	10,06%	0,0204	2,31%
Methods and instruments	0,016	1,80%	0,0018	0,20%
Standards and laws	0,0298	3,35%	0,0128	1,45%
Dependencies on own domain	0,0699	7,85%	0,0244	2,77%

	Social software system		**Project management system**	
	Index	**%**	**Index**	**%**
Number of cooperation partners	0,0639	7,19%	0,0693	7,76%
Location	0,0559	6,29%	0,0203	2,27%
Skill level in the agreed language	0,0089	1,00%	0,0232	2,60%
Type of engineering process	0,0312	3,51%	0,0022	0,25%
Organisation and company's culture	0,0065	0,73%	0,0034	0,38%
Size of the organisation	0,1401	15,76%	0,0104	1,16%
Intensity of collaboration	0,0077	0,87%	0,0117	1,31%
Distribution model	0,0698	7,85%	0,0119	1,33%
Number of interfaces	0,0171	1,92%	0,008	0,90%

Access to data	0,0394	4,43%	0,0416	4,66%
Skill level in Ithe use of IT	0,1425	16,03%	0,4454	49,88%
Influence of time	0,2275	25,58%	0,0117	1,31%
Vocabulary	0,033	3,71%	0,1216	13,62%
Methods and instruments	0,0077	0,87%	0,0291	3,26%
Standards and laws	0,0025	0,28%	0,0541	6,06%
Dependencies on own domain	0,0355	3,99%	0,029	3,25%

	Problem solving system		**Online translation**	
	Index	**%**	**Index**	**%**
Number of cooperation partners	0,0713	8,10%	0,0114	1,31%
Location	0,0614	6,97%	0,2439	28,00%
Skill level in the agreed language	0,108	12,26%	0,235	26,98%
Type of engineering process	0,0369	4,19%	0,0544	6,24%
Organisation and company's culture	0,0028	0,32%	0,0039	0,45%
Size of the organisation	0,0462	5,25%	0,096	11,02%
Intensity of collaboration	0,1776	20,17%	0,0334	3,83%
Distribution model	0,0059	0,67%	0,0136	1,56%
Number of interfaces	0,0261	2,96%	0,0088	1,01%
Access to data	0,0683	7,76%	0,0279	3,20%
Skill level in Ithe use of IT	0,0635	7,21%	0,0231	2,65%
Influence of time	0,0033	0,37%	0,0621	7,13%
Vocabulary	0,147	16,69%	0,0217	2,49%
Methods and instruments	0,018	2,04%	0,0025	0,29%
Standards and laws	0,00000051	0,00%	8,09E-07	0,00%
Dependencies on own domain	0,0444	5,04%	0,0334	3,83%

	Online conferencing		**Knowledge management system**	
	Index	**%**	**Index**	**%**
Number of cooperation partners	0,0117	1,34%	0,0484	5,38%
Location	0,2168	24,74%	0,0037	0,41%
Skill level in the agreed language	0,183	20,89%	0,0139	1,54%
Type of engineering process	0,0172	1,96%	0,00000663	0,00%
Organisation and company's culture	0,0895	10,21%	0,0052	0,58%
Size of the organisation	0,0111	1,27%	0,0189	2,10%
Intensity of collaboration	0,0416	4,75%	0,1009	11,21%
Distribution model	0,0235	2,68%	0,006	0,67%
Number of interfaces	0,0023	0,26%	0,0196	2,18%
Access to data	0,0086	0,98%	0,0442	4,91%
Skill level in Ithe use of IT	0,1704	19,45%	0,3538	39,30%

Influence of time	0,0191	2,18%	0,173	19,22%
Vocabulary	0,0214	2,44%	0,0097	1,08%
Methods and instruments	3,03E-07	0,00%	0,0077	0,86%
Standards and laws	0,0213	2,43%	0,0077	0,86%
Dependencies on own domain	0,0387	4,42%	0,0876	9,73%

	Instant messaging		**Forum**	
	Index	**%**	**Index**	**%**
Number of cooperation partners	0,0017	0,19%	0,3384	38,58%
Location	0,0508	5,79%	0,0888	10,12%
Skill level in the agreed language	0,154	17,56%	0,0659	7,51%
Type of engineering process	0,0531	6,05%	0,00000678	0,00%
Organisation and company's culture	0,0068	0,78%	0,0011	0,13%
Size of the organisation	0,0506	5,77%	0,0225	2,56%
Intensity of collaboration	0,1511	17,23%	0,0139	1,58%
Distribution model	0,0325	3,71%	0,0022	0,25%
Number of interfaces	0,0281	3,20%	0,0622	7,09%
Access to data	0,0052	0,59%	0,008	0,91%
Skill level in Ithe use of IT	0,0541	6,17%	0,0938	10,69%
Influence of time	0,0274	3,12%	0,0497	5,67%
Vocabulary	0,1358	15,48%	0,0197	2,25%
Methods and instruments	0,035	3,99%	0,0355	4,05%
Standards and laws	0,0275	3,14%	0,0022	0,25%
Dependencies on own domain	0,0634	7,23%	0,0733	8,36%

	E-Mail system		**Electronic meeting system**	
	Index	**%**	**Index**	**%**
Number of cooperation partners	0,0022	0,25%	0,0472	5,35%
Location	0,0615	7,09%	0,1252	14,18%
Skill level in the agreed language	0,234	26,99%	0,0856	9,70%
Type of engineering process	0,00000197	0,00%	0,033	3,74%
Organisation and company's culture	0,0389	4,49%	0,035	3,97%
Size of the organisation	0,0127	1,46%	0,0459	5,20%
Intensity of collaboration	8,71E-07	0,00%	0,1163	13,18%
Distribution model	1,22E-07	0,00%	0,0257	2,91%
Number of interfaces	0,2122	24,48%	0,0079	0,89%
Access to data	0,0042	0,48%	0,0115	1,30%
Skill level in Ithe use of IT	0,0477	5,50%	0,2252	25,51%
Influence of time	0,0027	0,31%	0,0055	0,62%
Vocabulary	0,2415	27,85%	0,0532	6,03%

Methods and instruments	0,0047	0,54%	0,0073	0,83%
Standards and laws	7,23E-08	0,00%	0,0291	3,30%
Dependencies on own domain	0,0047	0,54%	0,0291	3,30%

	Electronic calendar		**Document management system**	
	Index	**%**	**Index**	**%**
Number of cooperation partners	0,0352	3,97%	0,1089	12,20%
Location	1,11E-07	0,00%	0,02	2,24%
Skill level in the agreed language	0,0072	0,81%	0,0066	0,74%
Type of engineering process	0,0397	4,48%	0,0115	1,29%
Organisation and company's culture	0,0268	3,03%	1,44E-07	0,00%
Size of the organisation	0,0861	9,72%	0,0209	2,34%
Intensity of collaboration	0,0801	9,04%	0,0764	8,56%
Distribution model	0,0073	0,82%	0,00000062	0,00%
Number of interfaces	0,0063	0,71%	0,0103	1,15%
Access to data	0,257	29,01%	0,0756	8,47%
Skill level in Ithe use of IT	0,1264	14,27%	0,3201	35,86%
Influence of time	0,0128	1,44%	0,0529	5,93%
Vocabulary	0,1101	12,43%	0,0641	7,18%
Methods and instruments	0,0394	4,45%	0,0286	3,20%
Standards and laws	0,0515	5,81%	0,0116	1,30%
Dependencies on own domain	0,0000291	0,00%	0,0852	9,54%

	Discipline specific tool		**Co-Author system**	
	Index	**%**	**Index**	**%**
Number of cooperation partners	0,0105	1,19%	0,0136	1,53%
Location	0,0019	0,21%	0,0789	8,90%
Skill level in the agreed language	0,0092	1,04%	0,055	6,20%
Type of engineering process	0,0039	0,44%	0,0438	4,94%
Organisation and company's culture	0,0034	0,38%	0,0218	2,46%
Size of the organisation	0,0034	0,38%	0,0678	7,65%
Intensity of collaboration	0,0117	1,32%	0,1379	15,56%
Distribution model	0,078	8,80%	0,0066	0,74%
Number of interfaces	0,0347	3,92%	0,0113	1,27%
Access to data	0,0767	8,66%	0,0117	1,32%
Skill level in Ithe use of IT	0,2075	23,42%	0,1121	12,65%
Influence of time	0,0038	0,43%	0,0485	5,47%

Vocabulary	0,1972	22,26%	0,0704	7,94%
Methods and instruments	0,096	10,84%	0,0022	0,25%
Standards and laws	0,0616	6,95%	4,24E-07	0,00%
Dependencies on own domain	0,0865	9,76%	0,2048	23,10%

	Chat room		**Change management tool**	
	Index	**%**	**Index**	**%**
Number of cooperation partners	0,0015	0,17%	0,0732	8,18%
Location	0,0825	9,42%	0,0078	0,87%
Skill level in the agreed language	0,1657	18,92%	0,0036	0,40%
Type of engineering process	0,0546	6,23%	0,0019	0,21%
Organisation and company's culture	0,0122	1,39%	0,0064	0,72%
Size of the organisation	2,48E-07	0,00%	0,0121	1,35%
Intensity of collaboration	0,1851	21,13%	0,0056	0,63%
Distribution model	0,0284	3,24%	0,0371	4,15%
Number of interfaces	0,0029	0,33%	0,0092	1,03%
Access to data	0,0088	1,00%	0,0227	2,54%
Skill level in Ithe use of IT	0,038	4,34%	0,2915	32,59%
Influence of time	0,0637	7,27%	0,06	6,71%
Vocabulary	0,0826	9,43%	0,1108	12,39%
Methods and instruments	0,0014	0,16%	0,0602	6,73%
Standards and laws	0,0224	2,56%	0,1095	12,24%
Dependencies on own domain	0,1261	14,40%	0,0829	9,27%