Method Rationale Revealed
To Dante
Method Rationale Revealed – Communication of Knowledge in Systems Development Methods
Abstract

The practice of developing information systems with the support of information systems development methods is not new. A vast number of systems development methods have been suggested over the years in an attempt to solve the problems a development organisation might encounter. From early approaches like the Waterfall model to more modern monolithic methods such as the Rational Unified Process and the newest approaches exemplified in the Agile methods, the ambition has often been to find the silver bullet and the most effective ways to produce quality systems.

Methods are prescriptive by nature as they suggest action and as such they represent rationale. Thus, one can speak of a method rationale as the dimension within methods that motivate their existence. Method rationale is understood as the goal and value rational relations between a method’s underlying philosophy and its proposed actions.

During the methods’ evolution, the practice of systems development and the supporting systems development methods have been subjected to research from many perspectives. One possible way to understand the nature of the existing research is to separate it into two fields. The suggested fields have different strengths and weaknesses. The field of traditional research on information systems development (ISD) emphasise relevance in their studies but often overlook aspects of generalisation. The field of method engineering (ME) is highly formalistic and emphasise rigour but often miss aspects concerning relevance, such as the role methods play in peoples daily systems development efforts. In this dissertation, a polarisation of existing systems development method research is suggested in order to find a synthesis more capable of serving as a common ground for method research and for the understanding of the systems development method phenomenon. This is achieved through a proposed extension of the field of ME into the field of extended method engineering (EME).

The foundation of the EME is found in the concept of method rationale and a method component concept design capable of carrying and expressing method rationale. The method component concept design is applied, evaluated, and re-designed in three different empirical settings in order to ascertain its practical potential and the benefits in explicating the dimension of method rationale.

Keywords: systems development, systems development methods, method rationale, method components, method engineering.
Acknowledgments

Finally here… At a time like this, a lot of thoughts appear in my somewhat exhausted mind. It sure has been a long and an eventful journey and I am in debt to a number of people who have helped me during the entire process.

First of all, I would like to thank Åke Grönlund; professor, main supervisor and coach. Thank you for giving me the opportunity to write this dissertation and supporting me all the way!

My gratitude also extends to Fredrik Karlsson; co-supervisor, friend and wing-man in a lot of projects. This dissertation would never have been finished if it was not for your support!

Pär J. Ågerfalk; co-supervisor, Rolex-buddy and friend. We are there now! The next generation of the three Amigos! Or is it three Stooges…?

Over the years chapters, papers, drafts, and other ideas of mine have been scrutinised by others in order to find areas where my work could be improved. Thank you Göran Goldkuhl for early inspiration and for your input during my final seminar! Thank you Brian Fitzgerald and Erik Stolterman for discussions about general ideas in my dissertation. Karin Hedström, Anders Avdic, Johan Petersson, and Sattar Al-Haydar for reviewing what I had accomplished at the time of my half-time seminar. Thanks a lot! All of the persons mentioned above have given me valuable input and, when necessary, sent me back into a state of despair ;-)

A lot of colleagues have contributed to this work by just being there. I really can not think of a better group of co-workers! Johan Aderud; best friend and main co-conspirator to most of the mistakes I make socially, Ella Kolkowska; provider of polish sausage and Klubowe, Andreas Ask (the tough-looking soft guy), Mathias Hattaka (who showed me where the pikes are and how to collect seaweed), Annika Andersson, Jenny Lagsten and Emma Eliason; a trio of blondes who prove that what some say about blondes and intelligence is completely wrong. Blondes might have more fun though… Kenneth Åhlgren, this is really your fault altogether! Jens Axelsson, for being who you are.

My gratitude also extends to other people close to me: Mårten Magnusson; homeboy and spiritual brother, Catarina and Marcus, the Matsubaras and the Wistrands; a better family can not be imagined, Rammstein; for much needed ambience, Dexter Morgan and Jack Bauer for opportunities to think about something else for a change.

My deepest gratitude however goes to my immediate family. Jesper and Felix; the best bonus kids anyone could hope for, my loving wife Charlotte; everything I am, I am thanks to you! I love you! And finally Dante; my son and the light of my life, this dissertation is dedicated to you.

Kai Wistrand, Örebro march 2009
6. THE METHOD COMPONENT CONCEPT ................................................................. 97

6.1 Chapter Overview.................................................................................................. 97
6.2 Elicitation and Formulation of Design Goals for an Improved Method Concept ...... 98
6.3 Choice of Starting Method Module Concept......................................................... 100
  6.3.1 Analysis of the Method Fragment Concept ....................................................... 101
  6.3.2 Analysis of the Method Chunk Concept........................................................... 103
  6.3.3 Analysis of the Method Component Concept.................................................... 105
  6.3.4 Analysis of Requirements for the Starting Module Concept.............................. 108
    6.3.4.1 Conclusion Regarding Starting Method Module Concept.............................. 110
6.4 Conceptual Re-Modelling of the Method Component Concept............................. 112
  6.4.1 External View of a Method Component............................................................ 119
  6.4.2 The Interface and Content of a Method Component......................................... 120
6.5 Example of a Method Component ........................................................................ 120
  6.5.1 The Business Use Case Model Component Content........................................... 121
  6.5.2 The Business Use Case Model Component Interface ......................................... 124
  6.5.3 Connection of Method Components ................................................................. 127
6.6 Chapter Summary ................................................................................................ 129

7. APPLICATION OF METHOD COMPONENTS IN METHOD RECONSTRUCTION .......... 131

7.1 Chapter Overview ................................................................................................ 131
7.2 Case Setting ....................................................................................................... 132
    7.2.1 The PSU Project Assessment ........................................................................... 132
7.2 Method Reconstruction using Method Components ........................................... 134
    7.2.1 Method Reconstruction in the PSU Project ....................................................... 136
      7.2.1.1 Step 1 ........................................................................................................... 136
      7.2.1.2 Step 2 ........................................................................................................... 137
      7.2.1.3 Step 3 ........................................................................................................... 139
7.3 Conceptual Re-Modelling of the Method Component Concept ......................... 141
7.4 Method Rationale - Lessons Learned ................................................................. 146
7.5 Concluding discussion ....................................................................................... 147

8. METHOD RATIONALE IN PLAY – A SWEDISH ARMED FORCES CASE STUDY......... 149

8.1 Chapter Overview ................................................................................................ 149
8.2 Case Setting ....................................................................................................... 150
    8.2.1 The Modelling Task ....................................................................................... 151
8.3 Interpretative Research Approach ........................................................................ 152
    8.3.1 Data Collection and Analysis ......................................................................... 153
8.4 Rationality in Play ............................................................................................... 154
    8.4.1 Method Rationale as Attention Director ........................................................ 155
    8.4.2 Method Rationale as Consensus Builder ........................................................ 158
    8.4.3 Method Rationale as Check Point .................................................................. 160
8.5 Method Rationale – Lessons Learned ................................................................. 163
8.6 Concluding Discussion ....................................................................................... 164

9. METHOD RATIONALE AND METHOD COMPONENTS IN THE CONTEXT OF TEACHING
   INFORMATION SYSTEMS DEVELOPMENT METHODS ........................................... 167

9.1 Chapter Overview ............................................................................................... 169
9.2 Case Setting ....................................................................................................... 170
    9.2.1 Maturity in the Group .................................................................................... 171
    9.2.2 The Systems Development Method VIBA/SIMM ......................................... 171
9.3 Interpretative Research Approach ...................................................................... 172
9.4 Lectures ............................................................................................................. 173
    9.4.1 G1 Lectures .................................................................................................... 174
    9.4.2 G2 Lectures .................................................................................................... 176
PART I – Raising Questions

_There are two ways of constructing a software design; one way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult._

_C. A. R. Hoare_
1. Introduction

The need for method support for information systems development is not a new issue. It has been discussed since the 1960’s and several solutions and methods have been proposed. As a result of the expansion of the software industry these methods have become increasingly more complex. Today there is a plethora of available methods to choose from (Avison and Fitzgerald, 2003). These range from highly structured development methods such as the Rational Unified Process (RUP) (Kruchten, 1999) to less formalised methods focusing on speed and flexibility, such as extreme Programming (XP) (Beck, 2000). A method such as the RUP has great emphasis on supporting tools and production of documents as a means to generate support. For instance, RUP comes with a variety of tools for different purposes and more than 80 different major artefacts. The agile methods (of which XP can be regarded as one) also have a devoted following. These types of methods have a foundation in programming practise and try to capture best practises concerning systems development into light weight, or Agile, methods. The goal is to produce quality systems with less effort and documentation (Beck, 2000). Exactly how many methods available for use is not known, but Jayaratna estimated as early as 1994, that the number exceeded a 1000 available methods (Jayaratna, 1994). However the process of choosing a specific method and actually being able to deploy it in a development organisation can be a cumbersome process (Avison and Fitzgerald, 2003).

Today, research has shown that developers rather learn how to fail by adhering to methods than how to improve the quality of their processes. Using a method does not always come easy and it is difficult to comprehend the relation between project results and the method itself (Lyytinen and Robey, 1999). In this context it is understandable that organisations experience problems when it comes to method issues. Some organisations choose to handle the problem by not using any method support at all (Avison and Fitzgerald, 2003; Iivari and Maansaari, 1998). In those cases, some kind of tacit knowledge is usually the basis for the actions taken during development. Another way to put it would be to say that in such situations, there is, at best, an implicit method base, or at worst, pure guesses or ad-hoc based reasoning. In those cases one can argue that there is a clear lack of possibilities to control the events during development due to absence of the support usually provided by methods. Some organisations claim to use a specific method so that they appear more competent even though they still work the way they are used to (Parnas and Clements, 1986). Problems related to lack of productivity can always, in some sense, be handled from a methodological standpoint. By this, I do not mean that it is always the case that a new, well deployed systems development method is the silver bullet for every organisation. Rather, that it is probable that a well defined and understood information systems development method could help a development team to better pinpoint possible problems and areas of improvement. It could serve as a
common ground and a starting point when it comes to communication concerning the actual development process and how it could be improved. The role of the systems development method in these situations is as a medium for communication and a means to create synergy within a team.

The method, as such, can be regarded as a theory about actions to perform in order to reach a given goal or goals. Firstly, this tells us that we ought to treat systems development methods as theories (and not given truths) and treat them with the same scrutiny as other theories. Secondly, this implies that there is a conception of a target state. A goal, towards the method aims to guide you. Thirdly, there are propositions about how you should act in order to reach the desired target state. Another way to put it would be to say that there are relations (Ågerfalk and Åhlgren, 1999) between this target state and the proposed actions. Ultimately methods can be viewed as reason based statements regarding target states, and as such they represent rationality (Schluchter, 1981; Klein and Hirschheim, 1991).

The arguments concerning the knowledge bearing core of method is not particularly difficult to defend. Almost anyone would agree that a method’s role, in its simplest form, is to communicate knowledge from one person to another (Goldkuhl, 1999). Therefore, the conception that a method represents knowledge can be regarded as relatively clear cut. However, when looking at specific instances of methods one can see that it varies how well the rationality dimension is expressed and communicated. Some externalised methods describe, fairly well, the underlying arguments and reasons for the methods appearance. Others may include only a description of the process or the target state (Ågerfalk and Wistrand, 2003). Subsequently, since the methods express some kind of rationality we can speak of a method rationale which carries the underlying arguments for the methods appearance.

1.1 Information Systems Development Methods from a Historical Perspective
According to Fitzgerald et al. (2002) the history of systems development methods has its origins in problems identified in the early phases of computing. During the 1950s to the 1970s the task of producing applications for computers was considered to concern solving technical problems and programming. The earliest computers were used for scientific calculation and the problems the programmers experienced were related to this reality. The usage of computers has since then spread to involve almost every aspect of society today. During the period spanning from the early attempts of computing to today’s highly technical society, software and information systems have been produced to satisfy customer needs. However, the approaches for how to produce computer applications have evolved and changed over time.

Avison and Fitzgerald (2003) describes this evolution as consisting of four distinct eras with individual problems, challenges and solutions.
The first era they define is the *Pre-Methodology Era*. This era was the setting for the early days of computing. In reality going all the way back to the first computers but considerably more easy to define in terms of studying how information systems were produced during the 1960s and the 1970s. During this period, computers had spread beyond their previous military and scientific settings and were at that time becoming more common in other contexts such as banks, industries and other private or public businesses.

Consequently, new areas of application demanded new types of information systems. This, in turn required the persons responsible for programming these systems to learn how to produce information systems that met the customers’ needs. Traditionally, developers came from a computer technology background and had problems with understanding the businesses and the organisational contexts in which the systems would be implemented in. Information systems were often flawed and the development projects suffered from poor control and management. These problems and the state they resulted in are often referred to as the Software Crisis (Fitzgerald *et al.*, 2003).

As a way to better understand what the customers needed and to develop information systems that were more appropriate more efficiently, the idea of the Systems Development Life-Cycle (Canning, 1956) was introduced. This started the second or *Early Methodology Era* (Avison and Fitzgerald, 2003). In the late 1970s and early 1980s, systems development was characterised by approaches that divided the process of systems development into phases and stages. The idea was to ascertain that important development aspects were not omitted and to maintain a logical sequential order for how an information system should be developed. The systems development life-cycle or, more commonly, the waterfall model often suggested a series of phases including: planning or feasibility study, analysis, design, implementation, and maintenance (Fitzgerald *et al.*, 2002; Avison and Fitzgerald, 2003). Typically, one phase had to be completed before the next, hence waterfall. These leads to problems as information systems requirements tend to evolve and change. Although, there were ideas of iterations, these ideas were seldom recognized (Parnas and Clements, 1986; Avison and Fitzgerald, 2003). The phases were often associated to different techniques such as “flowcharting” (Avison and Fitzgerald, 2003).

The problems related to the Early Methodology Era can be summarized to regard: difficulties to meet the requirements, conservative systems design, instability, inflexibility, user dissatisfaction, poor documentation, and application backlogs (Avison and Fitzgerald, 2003). As a result, the developing industry moved into the *Methodology Era* where a number of methods were proposed in order to solve the earlier problems. According to Avison and Fitzgerald (2003) this era can be classified in to seven broad themes:

- Structured approaches – with a background in structured programming,
- Data orientation – entity relationship approaches with focus on storing data.
• Prototyping – the using of approximation of systems enabling user feedback prior to actual realisation.
• Object orientation – identification of objects, attributes and classes in order to handle reuse, inheritance and system behaviour.
• Participative – Strong emphasis on involvement of end users and other stakeholders.
• Strategic – Emphasis on an information systems strategy to support overall business objectives.
• Systems – A holistic view on businesses as constituted of human activity systems.

This lead to an abundance of proposed methods and the problems during this era often concerned how to choose “the best method” or, as methods became larger and more complex, how to adapt it for unique development organisations or projects. An even more drastic approach was suggested in order to find methods best suited for different needs. Kumar and Welke (1992) Odell (1996) and Brinkkemper (1996) suggest the notion of Method Engineering (ME) as a way to handle the problems of inadequate systems development methods. ME is founded in the idea that a new uniquely designed systems development method is assembled for each project out of a set of modular method constructs. Since then, ME has evolved into a research field of its own within traditional Information Systems Development (ISD) research. In comparison, ME research is often perceived as over-formalistic to other ISD researchers with a different research focus. Personally, I have experienced this at times during interaction with colleagues from other research institutions.

Avison and Fitzgerald continue with a description of the current era which they term The Post-Methodology Era (Avison and Fitzgerald, 2003). This era can be considered as a backlash against overconfident method use with the result that some developers have abandoned methods altogether. Those who still use systems development methods tend to abandon slavish adherence to method descriptions and seek new ways to develop systems by using contingency based approaches such as Multiview (Avison et al., 1998) or use Agile methods such as XP (Beck, 2000) or SCRUM (Schwaber and Beedle, 2002). Other possible alternatives during the Post Methodology Era are to develop systems with tools capable of generating code or simply outsourcing the development process or buying existing systems. Avison and Fitzgerald further conclude that even though systems development methods are seldom considered as fool-proof today they remain influential in the process of developing systems (Avison and Fitzgerald, 2003). The question today is rather to find a suitable way to develop systems by taking a balanced approach and adapt existing methods instead of looking for the perfect method like many developers did during the Methodology Era. This means that the field of ME still can produce results that can be considered as useful today. ME as a field is intended to construct unique
systems development methods, but adaptation of already existing systems development methods have also become more common. The need for adaptation of existing systems development methods with the help of ME have been called Method Configuration (MC) (Karlsson and Ågerfalk, 2004). With the advent of MC the field of ME have become more atone with the current view on systems development methods and their possibilities to contribute to the practice of developing systems. However, criticism can still be noticed from researchers that come from a more traditional information systems development (ISD) background (Mathiassen et al., 1996). Others have made a polarisation between different types of research concerning systems development methods. Ågerfalk and Fitzgerald (2005) choose to describe the two major research fields as method-in-action and ME. However, method-in-action oriented research, in their dichotomy, focus on the method usage situation as the method is enacted in development projects. There is a lot of research concerning systems development methods on a pure theoretical level or research that does not address usage situations. The proposed dichotomy excludes a lot of research on systems development methods. To describe research concerning systems development methods that does not apply an engineering perspective we need a broader concept focusing on more than just method-in-action. If ME can be defined as formalistic approaches to solve problems concerning systems development methods by rules and an intended logical coherence, ISD is defined in this dissertation as research on systems development and systems development methods that does not apply a formalistic perspective on the study objects. An easy distinction would be to say that ME research often have a tendency to aim for the rigorous, while ISD research rather emphasise relevance.

This calls for a distinction between ME and ISD for this dissertation. Thus, studies on how a systems development method is enacted in a project (method-in-action) can be considered part of the ISD field much in the same way as a study that addresses practical method usage or an ontological framework that classifies different typologies of systems development methods. Conversely, a study that focuses on how to define, store, and construct a systems development method will be considered to have a formalistic perspective and will be considered to be ME. Subsequently, in this dissertation I will refer to all research on systems development methods that does not apply a formalistic approach as ISD.

1.1.1 A Dialectic View on ISD Research

I have already argued for the possibilities to discern two different research agendas within the total field of ISD research. Research coming from the ME tradition with its formalistic approaches are often very oriented towards practice as they propose different ways for persons responsible for method management to construct unique systems development methods for their perceived needs (Brinkkemper et al., 1999; Rolland et al., 1999). Their research products can be
considered to be prescriptive to a higher degree than traditional ISD research products which may be in the form of a descriptive framework or lessons learned from systems development endeavours (Lyttinen, 1986; Fitzgerald et al., 2003). A lot of ISD research is devoted to criticising the assumption that usage of systems development methods would solve every problem a development organisation could have. They much rather emphasise the role of the individual developer and the tacit knowledge s/he possess. Introna and Whitley openly criticise what they call Method-ism (1997) and the dangers of singling out methods as the silver bullet. Their arguments can also be applied to criticise the method oriented, formalistic propositions coming from the ME field since ME have to assume that the solutions they propose also has to be followed more or less slavishly. They also have to assume that the constructed systems development methods are actually being used as intended and not altered while enacted. It is not easy to find examples of ME research where alterations of a systems development method during use are studied. Rossi et al. (2004) can be considered an example of this but apart from this example, I have not been able to find any others.

Method rationale is a central concept in this dissertation and it is possible to apply a dialectical perspective (Hegel et al., 2008) on how the notion of the knowledge bearing core of systems development methods have been treated in ISD and ME research. This view is also applicable when it comes to comparing other research coming from these two research fields. Of course, some researchers have produced research that can be classified as ME, but also published work that can be regarded as belonging to the ISD tradition.

A challenge the fields of ME and ISD ought to face is how these two fields compare in terms of their strengths and weaknesses. At times, they can be perceived as completely incommensurable. However, I do not believe that they are incommensurable as I see potential in searching for a common ground. By comparing the strengths and weaknesses it should be possible to find areas where these types of research can benefit from a synthesis. The differences between ME and other ISD research identify challenges concerning how to combine the knowledge produced in these two fields of research. The potential in combining this knowledge in a synthesis would possibly break new ground in research concerning systems development methods and practice. Below, I will argue for a perspective on systems development research from a dialectic perspective and also elaborate on a possible synthesis with a starting point in the concept of method rationale.

1.1.1.1 ISD as a Thesis
The main part of research on systems development and systems development methods has been conducted in, what I refer to as, traditional ISD research. This type of research focuses on various theoretical and practical aspects concerning the process of developing systems and the methods used as support for these
processes. ISD research as a thesis, as presented briefly here, will however emphasise aspects that can be related to the concept of method rationale.

ISD research can capture studies on methods on a purely theoretical abstraction level and studies which focus the political roles methods play in organisations (Introna and Whitley, 1997; Avison and Fitzgerald, 2003; Fitzgerald, 1996; 1997; Klein and Hirschheim, 1991; Russo and Stolterman, 1997; Hirschheim and Klein, 1991).

As ISD research address theoretical issues, the knowledge dimension of systems development methods have been explored before. Klein and Hirschheim (1991) elaborate on the rationality concept in relation to systems development methods and state that little research has been done on the idea that methods are based on social action and that this action has its foundation in rationality. They explore how four different rationality concepts are implemented in seven systems development methods: formal, substantive, communicative, and emancipatory. The argument is that there is a need to continue to explore how the different types of rationality can be better utilised and presented.

Russo and Stolterman (1997) draw on the result of Klein and Hirschheim and speaks of a rationality concept that can be divided in two, namely public and private rationality. The public rationality is expressed through a method and the private is inherent within a designer or a method user. In the ideal situation the two rationalities coincide and a state of rationality resonance occurs. This state means that the method user understands the method s/he is using can contribute to the same goals s/he desires. However, very little advice is presented on how one should try to facilitate the possibilities to reach rationality resonance. They merely establish the fact that it is difficult for the method user since methods not always present their rationality in a way that the method user can comprehend, and that the private rationality often is hard to explore since it can be regarded as tacit.

Yet another part of this community is oriented towards understanding of methods as a concept and method usage. Productions of different frameworks to clarify distinctions between different methods are proposed (Fitzgerald, 1998b; Iivari et al., 2000; Hirschheim and Klein, 1991; Hirschheim et al., 1997; Hirschheim and Klein, 1989; Hirschheim et al., 1996; Lyytinen, 1986; Lyytinen, 1987).

1.1.1.2 Method Engineering as an Anti-Thesis

According to Avison and Fitzgerald (2003), ME was proposed as a solution for problems related to inadequate systems development methods during the Methodology Era. Kumar and Welke defined ME (1992) as the discipline to analyse, construct and deploy methods that are adapted for different needs, such as a specific organisation or project. ME research is mainly solution oriented and aims to solve perceived method problems in organisations (Kumar and
Welles, 1992; Brinkkemper, 1996; Brinkkemper et al., 1999; Rolland and Prakash, 1996; Rossi et al., 2004).

However, the idea that systems development methods have a knowledge bearing core in the form of method rationale is not confined to traditional ISD. Originally the term method rationale came from the method engineering field (Oinas-Kukkonen, 1996). It is an elaboration of the more commonly known concept of design rationale where the records of the decisions during a design process are captured, maintained and re-used. These records capture why designers have made the decisions they have made during a design process. Why some specific functionality was implemented and some other was discarded (Conklin and Begeman, 1988). The decisions of this type are applicable in a systems development method perspective as well. This is shown in (Rossi et al., 2004) where the concept of method rationale is used to capture decisions regarding method evolution during use in projects over time. Capturing the knowledge regarding the configuration decisions can help an organisation to prepare new employees and facilitate their possibilities to understand how systems are developed in the company. The method rationale focused in those cases is called method use rationale and describes why certain types of method parts are or are not used in practise. Method construction rationale is the other type of method rationale and is an explanation why certain method parts are included in the original constructed method. So far, no attempts have been made to extract the commonalities of these two method rationale conceptions into common denominators and how the concept of method rationale can be utilised in a broader method application or communication perspective.

One can also argue that Situational Method Engineering (Brinkkemper et al., 1999) has a foundation in ideas of method construction rationale. This strategy involves method assembly using method fragments to address the requirements raised by unique development projects. The strategy follows the ideas of methods representing knowledge of how a target state can be reached and an analysis is performed to single out which needs should be fulfilled. They show how a systems development method can be built using modular method construction (Odell, 1996). These ideas are more thoroughly explored in Harmsen (1997) where project situations are characterised along with method parts in order to choose which method parts to use in the creation of a project specific method. The scenario aspects are properties of different method parts with the ability to contribute to a project’s success. By characterising a project situation one can derive which situation factors demands which method parts. This is an example of how a conception of a target state is assessed in relation to a method’s possibilities to achieve that target state. It is a process of matching method parts to specific projects that has foundations in the idea of method rationale, although implicitly.
A more explicit use of the concept of method rationale can be found in Karlsson and Ågerfalk (2004) and their method configuration approach. They propose that the concept should be a fundamental part of the process of tailoring systems development methods. Their starting point is a single systems development method named the base method. By analysing the base method in relation to the project situation from a rationality perspective, choices are made concerning which parts of the base method should be altered and how. This process is supported by a computerised tool to administrate and tailor the various systems development methods (Karlsson, 2005).

Another example of method engineering research with relations to the concept of method rationale can be found in for example Ralyté et al. (2003) where the concept of intentions is used to explain motives for choosing a certain method part in a method engineering project. However, they do not use the concept in relation to a target state, but rather say that an intention is a goal that can be achieved through performance of a process and that it refers to a task or activity in the process and expressed on an intentional level (Ralyté, 2002). A problem with this definition is that an intention only points to a specific task or activity and not to the desired target state, i.e. what would be achieved by performing that particular task or activity.

1.1.1.3 Extended Method Engineering - A Synthesis

Both fields present valid research on how method problems can be perceived and solved, however they rarely build on the total accumulation of method knowledge the both fields represent. This means that there are problems in combining research results coming from either tradition. ME is considered a subset of the accumulation of ISD research. However, this does not mean that the knowledge produced in these two traditions usually build on each other. On the contrary, they are often perceived as incommensurable.

The ME research produces methods aimed at solving specific or general problems. Their ambition is to facilitate the method users’ situation through development of meta methods (methods that operates on other methods rather than businesses) which, from an engineering perspective, can guide the users to well informed decisions concerning how the systems development method should be treated and applied. What I refer to as traditional ISD research, on the other hand, often brings insight through addressing inherent problems in method usage situations and problems in the way we perceive and value methods. They rarely present concrete solutions for the problems they identify, instead they often call for more research to improve understanding. Successful method cases in the field of ISD often have the form of lessons learned or success stories and there is little attempt to claim generality (Mathiassen, 2002; Fitzgerald et al., 2003). In comparison, ME research aims at general solutions and produces results that can help us re-design methods to a more appropriate version, or help us construct a completely new method. They, however, rarely consider the vast
knowledge of the systems development method as a social and political phenomenon.

By applying a dialectic perspective on ISD research and treating ME as an anti-thesis it is possible to formulate a notion of a possible synthesis that could create better synergies between systems development research and a different understanding of how systems development methods can be treated in practical situations. The idea of a possible synthesis stems from my observation of research results and the solution they presents. Henceforth, I will treat ISD and ME as two views on research concerning systems development methods and as such they represent two different fields of research. ME is, of course in reality a subset of ISD but the distinction is necessary in order to find possible areas where there are potential synergies that can be achieved through a synthesis. A starting point and my basic assumption in this dissertation, regarding research on systems development methods, is that the two fields could benefit from studying the results from each other more closely. As I see it, the differences between ISD and ME present a challenge aimed at researchers studying systems development methods. The challenge consists of finding a common ground where the philosophical and interpretative perspectives of ISD can draw on the knowledge concerning method representation and construction produced in ME research. At the same time, there is a challenge in finding ways for how to incorporate systems development methods’ philosophical dimension into the field of ME. There is a need to explore how the two fields can contribute to develop new theories, solutions, and frameworks by taking each other into account.

![Figure 1. Main focus for ISD and ME research](image)

Figure 1 above shows the different main focus for ISD and ME respectively. There, I have compiled a number of types of perspectives on systems development method research taken within the two research fields.

A synthesis aimed to meet these challenges has a possibility to create a common ground for systems development method research that can combine the rigour of ME without compromising with the relevance provided by ISD. The ambition is to build a bridge of common understanding between the two fields.
of ISD and ME that could result in enhanced understanding of the systems development method phenomenon and possible forms of its representation. Hopefully, a synthesis can be found useful from both perspectives as well as in actual practices. In the dissertation this will be done by exploring the core of the methods knowledge dimension, i.e. the concept of method rationale. Bringing these elements into the light and defining what they are, and how they can be used would bring clarity into the diverse field of systems development method research. A clear definition of this concept and explication of its fundamental ontological elements could serve as a tool for communication and a solution space between the two fields.

The proposed synthesis will be a suggestion for how the systems development method phenomenon can be understood and used in research and practice. Since the synthesis, by definition, aims to provide something new it will necessary to name the synthesis. Otherwise, it will be difficult to denote and talk about it. ME is, as already stated, considered to be a research field within the larger field of ISD research, and here I mean ISD in its wide term. Practically, it would be difficult to argue for a synthesis that would call for a re-definition of all ISD research, it would be more feasible to suggest a synthesis that calls for a development of the field of ME as it has a common formalistic foundation aiming for rigour. This ME foundation is more easily discernable than a common denominator in the field defined as ISD in this dissertation. ISD, in the sense I refer to it in this work is rather defined by what it is not (formalistic, highly descriptive approaches to construct or adapt systems development methods) and I have chosen to classify all research on systems development methods that is not ME as ISD. My proposed synthesis should subsequently involve a re-definition or enhancement of something that is possible identify easily. For the above stated reasons, I have decided to take a starting point in the field of ME for a description of my synthesis. Subsequently, my synthesis will be referred to as Extended Method Engineering (EME).

I will use the formalisation tradition from ME in order conceive method concepts that also takes the rationality dimension of methods into consideration. By doing this I have a possibility to create a concept that has a chance to bring in ISD knowledge in the field of ME and expand the boundaries concerning what kind of research could be considered ME.

1.2 Research Questions

When it comes to conducted research on various aspects of systems development methods one can discern two major method research agendas or fields (Ågerfalk and Fitzgerald, 2005). Both fields address, directly or indirectly, the concept of method rationale. The notion of viewing research about systems development methods as polarised is therefore not new. The problems connected
with this polarisation concerns the apparent reluctance to draw on results coming from the “opposing” field.

In order to end up with a synthesis in the form of EME, there are a number of aspects that need to be clarified. First of all, I must define the nature of the field of EME in relation to the suggested polarised fields and explain its benefits. Secondly, I must explore the nature of the rationale dimension of systems development methods and define what method rationale is and how it can be used in EME research and the benefits of applying a method rationale perspective in practical method usage situations. Traditionally, in ME, research has been about finding new forms of representation of methods and the question of how to use these in order to construct new methods or to adapt already existing ones. Since this dissertation is considered to belong in the ME tradition, it will also contain discussions about, and definitions of, new representation forms. The ambition is to create a formalised method concept capable of carrying and expressing method rationale. This means that I will focus on the communicative aspects of systems development methods and method rationale in various empirical settings. The object of study during the empirical case studies will be communication. Communication is defined as a process by which we assign and convey meaning in an attempt to create shared understanding (Fiske, 1994). This means that I will study how methods are taught or presented in order to communicate certain ideas and results. Examples of these results could be an analysis, or how things have been agreed upon.

The concept of method rationale also needs to have a foundation in empirical observations. This means the concept must be defined in a way that practitioners can appreciate and also in a way that enables tests of developed frameworks and theories. Basically, this means that method rationale as a concept needs to be studied and defined ontologically and empirically. Knowledge of this type could serve as examples of research in EME.

Based on the discussion above, my research question is:

How can method rationale be defined, represented and used in extended method engineering to improve communication about systems development methods?

1.3 Demarcations and Related Work

Traditional method engineering usually involves method construction or adaptation (Brinkkemper, 1996; Rolland and Prakash, 1996; Harmsen, 1997; Karlsson, 2005). These tasks have been extensively covered in ME research and I have no reason to conduct studies of similar nature. My interests rather lie in application of ME concepts and ideas in what normally is referred to as ISD. By doing this I move the boundaries of ME and argue for the need of a common ground in the synthesis of EME. This means that my theoretical work primarily will be of ME nature. I will discuss concepts such as method modules and
method rationale and formalise these in the ME tradition. The ISD field will play the role of the empirical arena for my case studies. I will bring my elicited ME concepts into this arena and evaluate the results. Thus, a demarcation can be found in the application of traditional ME concepts in ISD settings.

The concepts of method rationale and method components have been developed together with other researchers. It is therefore wise to add a note on related research and how these research products differ. The first definition of Method rationale was conceived together with Pär J. Ågerfalk and presented at ICEIS 2003 (Ågerfalk and Wistrand, 2003). The interpretation of the concept and how it is applied does not differ from how it is used early in this dissertation.

Method components and their role as containers of method rationale have been developed together with Fredrik Karlsson. In Karlsson’s dissertation Method Configuration : Method and Computerized Tool Support (Karlsson, 2005) the concepts of method rationale and method components play a central role. During the process of developing these concepts, myself and Karlsson have worked closely on the conceptual evolution and have jointly presented results from our endeavours (Karlsson and Wistrand, 2003; Wistrand and Karlsson, 2004; Karlsson and Wistrand, 2006). Even though the central concepts of method rationale and method components have been developed jointly, the use of these concepts in both our dissertations differ greatly. Karlsson have focused on traditional ME and suggested an alternative approach with the starting point in an already existing systems development method, referred to as the base method. The approach is called Method Configuration and aims to solve problems connected to adapting a large method such as the RUP for an organisation. Karlsson uses the method component and the method rationale concepts in a meta method called Method for Method Configuration (MMC) and a computerized tool (MC Sandbox) to support the MMC process.

Method components and method rationale have previously been proved successful in a traditional ME context. In this dissertation, the emphasis rather lies on the theoretical aspects of these concepts and explorations of how they can be applied in contexts not typically related to ME.

1.4 Purpose of the Study and Summary of Contributions

The idea of applying traditional ME concepts in a new setting and showing how these can be used to understand the practise of systems development is the primary contribution of this dissertation. The synthesis, in the form of EME aims to show the practical benefits of using a method component and method rationale perspective on practical method application situations. They are also examples of how research can be conducted with an EME perspective. This involves formulations of concepts that would help to extend the field of method engineering into new grounds and enable a new type of ME research that previously has not existed. This would facilitate knowledge transfer and allow
ME researchers to appreciate the results conceived by traditional ISD researchers and vice versa.

This will be achieved by a dialectical analysis of the suggested polarisation of systems development method research into two fields and a formulation of the field of EME combining the strengths of the two. The EME research field will aim to serve as a common ground for systems development method research and will be intended to have an appearance that would interest researchers coming from both traditional ME and traditional ISD.

The suggested field of EME will have a foundation in method rationale. In order to create such a foundation I will analyse the existing fields with respect to what is currently known about method rationale and formulate an extension of the ME field incorporating method rationale. This means that I will also have to define what method rationale is.

In order to create a common understanding of the basic concept of systems development methods, I will also design a method concept capable of carrying and expressing method rationale for EME research.

Finally, I will show examples of how EME research can be conducted by applying my concepts in qualitative case studies. This part of the dissertation aims to show that it is possible to combine the fields of ME and ISD in a way that previously has not been done, and ascertaining the viability of the suggested field of EME.

However, this dissertation does not only aim to produce contributions within the area of research. All research should aim to produce results that can be perceived as meaningful in practical settings. The application of traditional ME concepts in non-typical settings should therefore also be seen as examples where different types of understanding about systems development practise can be achieved. The empirical cases will explore the benefits of an explicit method rationale perspective and analyse how method rationale and method components can have a positive effect in practical systems development method settings.

My contributions can be summarised as:

- Formulation of the field of Extended Method Engineering
- Definition of method rationale
- Construction of a method concept capable of carrying and expressing method rationale
- Exploration of the potential of applying a method rationale and method component perspective in practical systems development method settings.

1.5 Target Groups

The intended target groups for this dissertation are mainly twofold. Researchers coming from the field of ME are one of the targets since the dissertation aims to extend that particular field. However, researchers coming from an ISD tradition
might very well be interested in reading about how ME concepts can be applied in qualitative research, typically belonging to the ISD field.

The second target group consists of people who teach methods or work within the practise of systems development methods. The empirical cases show examples for how a method rationale and method component perspective can help practise. This can be done by improved understanding of the roles method rationale and method components play in communication in practical situations.

The intended reader is familiar with the process of systems development and systems development methods. Knowledge of the field of ME is recommended, although not required.

1.6 Structure and Reading Instructions
The dissertation contains four parts. Below is an overview of the dissertation’s structure along with instructions for how the dissertation can be read, depending on interest.

PART I
This part contains two chapters:

1. Introduction
This part covers my initial research problems and argues for the need for a synthesis of the fields of ME and ISD. Here I also present my research questions and intended contributions. Any reader interested in what I find interesting and peculiar might want to start here.

2. Research Approach
Here I present my research strategy for the dissertation. It contains a description of how I have conducted my literature review which is the primary foundation for the method rationale definition and the formulation of the field of EME. It also covers discussions about how I have formulated design goals for my method concept and applied them. I also explain the design science research approach I have applied in the evolution of the concepts of method rationale and method components. The chapter ends with a description of the roles case studied play in my dissertation and how I have treated them.

This chapter is intended to give an overview of my entire research process. This is a chapter suitable for the reader interested in my use of research method and what the general structure of argumentation I am trying to create looks like.

PART II
This part contains two chapters of theoretical nature:

3. Systems Development Methods – A Multifaceted Concept
This chapter is intended to provide an orientation background to the concept of systems development methods. Much of the content in this dissertation involves
the concept of systems development methods and I found it fitting to devote one chapter to a presentation of how this concept can be understood and how it has been used in the fields of ME and ISD. The reader interested in an orientation about general systems development method representations and conceptualisations and how they have been used in previous research might want to read this chapter.

4. Method Rationale
In this chapter I elaborate on the concept of method rationale and end up with a definition. This definition is then applied in further chapters as input for design or as an analytical tool. This is also the chapter where I explore how method rationale has been used in the fields of ME and ISD. This results in a synthesised ontological framework for how method rationale can be understood in the context of EME.

As method rationale is a very central concept in this dissertation any reader who wants to understand how I have defined and used the concept should read this chapter as it explains the fundamentals of a central concept and defines the field of EME.

PART III
The third part contains five chapters and can be considered to be the part where my empirical results are presented, even though chapter 6 can be considered as more theoretical. All chapters are considered suitable reading for anyone who is interested in an exploration of the benefits of applying a method rationale perspective, method components, and how EME research can be conducted.

5. The Role of Method Rationale in Method Implementation
This chapter presents a case study at Posten IT where we followed the initial phases during an implementation of a systems development method. During this case study I look at the role of method rationale during these tasks and evaluate their current approach for method assessment and reconstruction. I also begin with a collection of requirements for a method concept more fitting for EME.

6. The Method Component Concept
In this chapter we conduct an analysis of the collected requirements from chapter 5 and formulate design goals for a method module concept for EME. We then compare three existing method concepts in an effort to choose a starting concept for a re-design process of a method module concept for EME. We ultimately chose the method component concept and re-designed it according to our elicited design goals.
7. Application of Method Components in Method Reconstruction
In this chapter we continue our research at Posten IT. The case study involves an application of the re-designed method component concept and presents an alternative approach for method reconstruction which is compared an evaluated with the approach presented in chapter 5.

8. Method Rationale in Play – A Swedish Armed Forces Case Study
In this chapter I conduct a case study during a series of modelling seminars in a military project. I focus on the communicative role method rationale can play during method-in-action.

9. Method Rationale and Method Components in the Context of Teaching
This final chapter of empirical nature reports on a case study experiment where I use method rationale and method components during teaching. The chapter aims to identify possible benefits of applying these perspectives during systems development method training.

PART IV
This part only covers one chapter:

10. Summary of Findings and Conclusions
In this chapter I revisit my theoretical and empirical results and formulate overall conclusions. These are related to my research question and I return to the notion of a synthesised filed of EME. The dissertation ends with some notes regarding possible future research. This chapter can be studied by a reader interested in a compilation of my research efforts and how these are related to my initial research questions.
2. Research Approach

2.1 Chapter Outline

This chapter follows on the problems identified in chapter one. I have identified a possible synthesis between the two major research fields focusing on system development methods, the field of method engineering (ME) and a field I choose to call information systems development (ISD). The synthesis concerns the fact that there are no real attempts to combine the research results the two fields generate. The field of ISD does not pick up on the solution oriented, problem solving and hands on techniques suggested from the field of ME. Conversely, the field of ME does not take philosophical, political and problem identifying advantages from the field of ISD. This dissertation will try to find a common ground of understanding which will facilitate knowledge transfer between ME and ISD. The starting point for this common ground must have a communication based foundation, and by this I refer to the ability to transfer knowledge from one field of research to another. The relevant knowledge about systems development methods, when it comes to communication, is called method rationale. A concept introduced in chapter one and a concept that will serve as a main thread throughout the dissertation.

Since both fields address the concept of systems development methods I find it fruitful to explore which aspects of the concept that can be used to facilitate this communication. This means identifying, exploring, designing and testing a systems development method concept that can carry method rationale and is viable and useful in both fields of research. This approach means that I will find common denominators in the systems development method concept so that researchers from both fields can talk about, and mean the same thing when they are talking about systems development methods and method rationale. Ultimately, this means that this dissertation is about finding a useful systems development method concept and to show how this concept can be used in both fields of research. Hopefully, this will give researchers the possibilities to test their ideas in another research field. I see no reason why results should be limited to one field of research when it seems possible to use them in another. My ambition is not to try to create a new field for the whole field of research concerning systems development methods but rather to enable fruitful impregnation of two diverse research fields.

The chapter is a description of how this task will be performed. It covers the whole research process from the initial literature review to the design science approach of a new systems development method concept and ultimately, tests of the new concept in various case studies. The case studies will test the new systems development method concept in both ME as well as ISD settings. The results of the tests will be used to evaluate the concept and ascertain whether or not it could be utilised in a capable, useful manner in EME.
2.2 Literature Review

Firstly, I will conduct an analysis of the existing research in the fields of ME and ISD and explore the concept of systems development methods and how it has been perceived earlier. The focus of my literature review will be research result that concerns the knowledge dimension of systems development methods. The knowledge dimension can be characterised as divided in two separate but intertwined perspectives. Firstly we have the perspective commonly used in papers of the ISD type. This perspective generally describes the systems development method from a philosophical angle, such as Hirschheim & Klein (1991) or Iivari et al (2000). The perspective can also address issues related to individual method use (Fitzgerald, 1997; 1998a; Livari and Maansaari, 1998) or considers political implications of systems development methods as they are often given the role of governing tools (Hirschheim et al., 1996; Hirschheim and Newman, 1991; Parnas and Clements, 1986; Beynon-Davies and Williams, 2003). As such they govern the systems development process and direct the method user towards certain and preconceived actions for a variety of reasons.

The second perspective which addresses the knowledge dimension is constituted upon the variety of research results addressing descriptions of what a systems development method is made of. These papers are common within the field of ME and often present ways to modularise a systems development method into smaller, more apprehendable pieces suited for the task of constructing or adapting a systems development method. Examples of these types of papers can be found in Brinkkemper (1996; 1999) Rolland and Prakash (1996) or Kumar and Welke (1992). The knowledge dimension is often implicitly addressed in these types of papers, though it is possible to find it if you analyse the way the authors talk about their individual systems development method concepts and modularisation principles, i.e. how they talk about important method parts.

These two perspectives will be my guiding light during my literature review. By reading a vast number of selected journal papers and conference proceedings, I will choose papers based on their abstracts. The selected papers will be studied more carefully so that I can decide whether or not the papers really address the knowledge dimension of systems development methods in a way that is useful for the purpose of this dissertation.
Table 1. Sources for the literature review

<table>
<thead>
<tr>
<th>Journals:</th>
<th>Conference proceedings:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information and Software Technology</td>
<td>International Conference on Information</td>
</tr>
<tr>
<td></td>
<td>Systems (ICIS)</td>
</tr>
<tr>
<td>Information Systems</td>
<td>Conference on Advanced Information Systems</td>
</tr>
<tr>
<td></td>
<td>Engineering (CAiSE)</td>
</tr>
<tr>
<td>Information Systems Frontiers</td>
<td>European Conference on Information Systems</td>
</tr>
<tr>
<td></td>
<td>(ECIS)</td>
</tr>
<tr>
<td>Information Technology and People</td>
<td>Hawaii International Conference on System</td>
</tr>
<tr>
<td></td>
<td>Sciences (HICSS)</td>
</tr>
<tr>
<td>Journal of Systems and Software</td>
<td>Information Systems Development (ISD)</td>
</tr>
<tr>
<td>Knowledge-Based Systems</td>
<td>International Workshop on Evaluation of</td>
</tr>
<tr>
<td></td>
<td>Modelling Methods in Systems Analysis and</td>
</tr>
<tr>
<td></td>
<td>Design (EMMSAD)</td>
</tr>
</tbody>
</table>

Requirements Engineering
Software Quality Journal
MIS Quarterly
Information Systems Journal
Informations Systems Research
ACM Computing Surveys
ACM Transactions on Information Systems
ACM Transactions on Software Engineering and Methodology
ACM Transactions on Database Systems
IEEE Transactions on Software Engineering
IEEE Software
European Journal of Information Systems
Scandinavian Journal of Information Systems

The identified papers, addressing the knowledge dimension of systems development methods will be read and analysed according to Webster and Watson’s (2002) approach for literature reviews. They propose a concept centric literature review which helps the authors to synthesise their results into something that extends existing research. My literature review will pick up on concepts identified in the two research fields focusing on method knowledge and the knowledge carrying parts of methods. Identifying the knowledge carrying parts is essential for the possibilities to design a new system development method concept that is capable of superseding the sometimes conflicting views on the topic that exist today. This means that my concept centric approach will have to focus on both fields and draw upon research results that are deemed possible to transfer between the fields of ME and ISD. The aim is to find ideas and theories about method knowledge that have previously been explored in one research field, but not the other. In the end of
the literature review a synthesis of ideas and concepts will be presented that can extend the existing research on method knowledge through a more elaborated and homogenous method rationale theory. In the end of the literature review I will argue for the synthesis’ possibilities to support future research in EME.

2.2.1 Elicitation of Design Goals

The analysis and synthesis ultimately focuses on elicitation of design goals for a systems development method concept that can express method knowledge and be useful in both fields of research. Based on the literature review and the following synthesis I will elicit requirements for the new method concept. These requirements will be reformulated as design goals for an improved systems development method concept capable of expressing method rationale in a way that is meaningful and useful for knowledge transfer between the field of ME and ISD. I will list the design goals as criteria that the new improved systems development method concept has to address. By analysing the results from the literature review in this fashion and drawing upon the resulting synthesis, these design goals will be the starting point for a design of the new concept.

In order to find the appropriate design goals, it is necessary to conduct analyses of existing conceptualisations of systems development methods. This not only shows weaknesses and strengths in the original concepts, it also gives input to defining a specified form of representation for my new concept. Traditionally, these types of ideas are more common in the field of ME, although examples can also be found within the field of ISD. Typical examples of research focusing on method representation within the ME field can be found in Brinkkemper et al. (1999), Harmsen (1997) or Ralytė et al. (2003). There

Figure 2. General model for the literature review
have also been attempts to conceptualise methods within the field of ISD but in comparison to the field of ME, which most commonly focuses on method construction, the conceptualisations in the field of ISD generally focuses ontological or political aspects of systems development methods. Examples of these latter conceptualisations can be found in Jayaratna (1994) Goldkuhl et al (1998), Cronholm and Ågerfalk (1999), Cockburn (2000) or Benyon-Davies and Williams (2003).

2.2.2 Design of a New Concept

The elicitation of design goals for the new systems development method concept will draw upon the synthesis from the previously conducted literature review. During this design phase I will use the design goals to create an improved concept. The design goals will function as guiding principles for my work which will ensure that the new improved system development concept will have the capability to fully address the problems identified during the literature review. This will be done through combination of the synthesis and the various ideas for method representation. The design goals will be formulated as properties of the new systems development method concept.

The properties will be defined and incorporated in a formal description of the concept, ensuring that it will have the capability to express method rationale in a way that can be perceived as useful in both fields of research. By using existing knowledge on method representation I will choose a purposeful shell of representation and extend it into a container for the derived properties of method rationale. If needed, I will also incorporate other ideas from already existing representation forms. The result of this process will yield a new systems development method concept in an iterative fashion through my case studies.

Figure 3. General model for the design phase
The design activities will be conducted with the help of conceptual modelling. By this I mean that I will focus on the constituents of method rationale and how this can be represented in a method concept. This involves definitions of how the identified concepts are associated and also possible properties of individual concepts and associations. The formalisation of a method concept capable of carrying and expressing method rationale requires a formal form of representation. I have chosen to apply UML (Booch et al., 1999) for the representation as this would allow storage and manipulation of the method concept as well as a conceptually stringent representation. The method rationale definition will also be described with the help of UML as it needs to be defined with the same level of detail as the method concept.

### 2.2.3 A Design Science Approach

My notion of a method module concept with the capability to carry and express method rationale is treated from a Design Science perspective. According to Hevner et al. (2004) the design science paradigm address questions such as “what is effective”, rather than questions of the type “what is true” which is more common within a behavioural science paradigm. My ambition to create a method concept and apply it in different settings corresponds to this notion. Particularly, since I will evaluate and re-design my method concept after each case study.

A process for applying an idea, formulation of a solution, develop a design and testing and evaluating the design is suggested by Pfeffers et al. (2006). Their proposed process contains 6 steps which are applicable for my research process. Below I will describe my research process in the context of their process.

The first step they suggest is **Problem identification and motivation**. In my research process this step corresponds to my initial research question and my literature review and my proposed synthesis of the fields of ISD and ME. Here I motivate the reasons for my study and propose a tentative solution. These tasks are performed in chapters 1 and 4. This step is followed by **Objectives of a solution**. During this step requirements are gathered in order to create a design that can solve the previously identified problems. In my dissertation this step corresponds to the parts where I elicit design goals for my method concept capable of carrying and expressing method rationale. These can be found in chapters 5 and 6. The next proposed step is **Design and development**. During this step the process of designing a solution is presented. The solution is a design intended to solve the problems identified during step one. In chapter 6 I present a re-designed method component concept capable of carrying and expressing method rationale as my primary design. The following step is **Demonstration**. During this step the design is applied and tested in various settings. In my case studies I apply the method component concept as well as the method rationale definition in order to ascertain their viability in theoretical and
practical systems development method settings. After each Demonstration step it is important to evaluate the results and compare them to the original design goals. This is done during the Evaluation step. After each case study in my dissertation I return to my previous requirements and see if I have learned things during the case study that would motivate a re-design. When needed, I make alterations to the method component concept and the method rationale definition. Examples of this step can be found in chapters 5, 7, 8, and 9. The final step suggested in Pfeffers et al. (2006) is Communication. During the overall research process papers have been presented in journals and at conferences giving opportunity to fellow researchers to scrutinise my progress so far. However, this dissertation can be considered to be the primary communication outlet of my research process.

2.3 Case Studies
In order to ascertain the level of viability of the systems development method concept it will be necessary to test it in various practical settings. The ideal way for testing a new or improved concept in my case is to explore its possibilities to fulfil the requirements elicited during the literature review. This will be done during a series of case studies (Yin, 1994). These will take a starting point in the requirements and explore whether the designed properties can deliver what they are designed to deliver, concerning usefulness in both fields and enabling communication of systems development method knowledge though method rationale. Most of these cases will take place in a setting involve practitioners of various kinds. Some cases will involve developers and some will involve method engineers. A method engineer is a person responsible for adapting or constructing a unique systems development method for an organisation or a specific project. Other cases will involve method users on various levels of competence. One case regards method communication and the possibilities to express method rationale in an academic setting where students will make acquaintances with a systems development method not earlier introduced to them. The ambition is to try to explore the viability of the systems development method concept in various practical settings to establish whether it truly can be used to communicate the ideas of method rationale in settings where ME concepts typically have not been applied. The strategy of conducting research in collaboration with practitioners has been successful earlier (Benbasat et al., 1987; Mathiassen, 2002), not least since it gives a natural and concrete point of measure.

If the practitioners find the results useful, then the obvious conclusion would be that they probably are, at least to some degree and hopefully general enough to make sense in a wider perspective. Since the topic of this dissertation is in the area of systems development and this is a practise which is conducted somewhat similarly over the world. Most projects tend to have some kind of requirements phase followed by a development phase and finally a delivery phase of some
sort. These phases can involve use of formalised systems development methods which would fit the scope of this dissertation. It is impossible for me to guarantee that the results ultimately presented in this dissertation will be deemed important by all practitioners, this fact is mostly connected to the level of method maturity in the organisation and knowledge of, and adherence to research results and ideas that come from the fields of ME and ISD.

The method specifics of each case study will be explained respectively in each case study chapter. All of the case studies will have a qualitative approach (Patton, 1990) to the evaluation. This means that I will not try to quantitatively measure the efficiency of the new systems development method concept, but rather analyse how it is perceived and whether the properties seem to yield the results they are intended to. The various evaluation case studies will follow the principles of qualitative case studies in accordance with Lee (1989). Exactly how will be given account for in each of the case study chapters.

2.3.1 Case Study Evaluation

During the various case studies the newly designed systems development method concept will be evaluated. The results of the evaluations will give feedback about whether the initial design goals were correct and whether the new concept can express method rationale in the intended way. Problems identified during the evaluation case studies can be considered flaws in the design or input for a review of the initial design goals. If needed, the systems development method concept will be re-designed in accordance with the sum of these results. Any attempts to redesign the concept will be elaborated in the dissertations conclusion chapter.

After each of the case studies I will collect the lessons learned concerning the concept of method rationale. If my case study experiences suggest that my definition needs alteration or evolvement they will be handled in the lessons learned sections.

Figure 4 describes the overall research process, covering the literature review as well as the case studies.

The figure also shows which activities are covered in which chapters. This is to enable the reader to understand where in the dissertation I present my results, and in what order.
According to Eisenhardt (1989) Theory can be used in three different ways in relation to case studies. Firstly, theory can serve as an initial guide to design and
data collection. In this way theory is used to give an initial focus during collection of data that are intended to evolve into some kind of theory. An example of such type of research can be found in, for example, Grounded theory (Strauss and Corbin, 1998). It also means that theory can function as a way to gather requirements for the design of something. In this latter interpretation, the application of a systems development method during analysis in order to create an information system would be considered an example of this type of theory use even though a development project is rarely considered and treated as a case study. Secondly, Theory can also serve as part of an iterative process of data collection and analysis. This type of use applies the conceived theory in a case study context. The theory directs the researchers attention towards certain phenomena, much in the same way a method does, and is tested though use. If needed the theory is evolved and applied in the same case study or another. The aim is of course to fine tune and the theory and test its viability empirically. Thirdly, theories can exist as the resulting final product of the research. By then the theory has evolved iteratively and tested empirically (Eisenhardt, 1989).

My case studies can all be categorised as belonging to Eisenhardt’s second use of theory as I use the concepts, frameworks, and definitions from PART II in three different settings; The Swedish Armed Forces, Posten IT’s RUP implementation project, and a university course in systems development methods. In these settings, the theoretical parts of this dissertation are used as a way to interpret the case studies and to test method concepts and approaches. During the case studies I evaluate my theoretical elements in relation to the empirical material. Thus, I have opportunities to let my concepts and definitions evolve if needed as a direct result from their application in practical research.

My final conclusions, in chapter 10 will include a description of the evolved theoretical elements and as such represent a theoretical foundation for method rationale and EME. Hence, chapter 10 can be regarded to follow Eisenhardt’s third use of theory.

2.4 Timeline and Output during the Research Process

During the course of my PhD project I have spent time on different theoretical problems and cases at different times. Figure 4 gives an overview of the entire research process and defines temporal aspects when it comes to the order of activities I have undertaken. However, it does not describe my PhD project in actual time and effort. It does not say anything about output in terms of conference and journal papers during the research project. Therefore I have a model describing an overview of my theoretical and empirical studies. This model can be found in Figure 5 and describes how what I have studied and when.

During the course of a PhD project a series of research papers and reports are written. Some are discussed in internal seminars and others are shared with others during conferences or in scientific journals. The numbers in the model are
intended to denote published output from my research. For instance, the “2” that can be found in the “Method Rationale”-field means that paper no 2 was concerning the concept of method rationale and published in the spring of 2003.

**List of Related Publications**


PART II – Exploring Theoretical Concepts

The mind is like a parachute. It doesn’t work unless it’s open.

Frank Zappa
3. Systems Development Methods – a Multifaceted Concept

An integral part of any research on a given topic is the very core of concepts that decide what the research really concerns. These core concepts function as a paradigmatic definition of how problems and solutions are perceived, solved and communicated. However, they do not only give guidance to how issues are to be treated. As a consequence they also delimit the possibilities to “think outside the box” since the concepts directs attention towards certain kinds of phenomena and, hence, also away from other kinds of phenomena. This chapter aims to present a necessary background on the topic of systems development methods.

The fields of ME and ISD have differences in how problems concerning systems development methods are treated and perceived. It is likely that their conceptualisations differ since they obviously perceive the realm of method use differently. This chapter is devoted to the clarification of how the concept of systems development methods have been treated in the research fields of ME and ISD respectively. I will show how fundamental differences in the way the concept is defined and used in the two fields and how they differ in terms of strengths and weaknesses and how these can be considered to be challenges for the field of systems development research.

Starting in a neutral corner, according to Wikipedia, a method is defined as “codified series of steps taken to complete a certain task or to reach a certain objective”. Going into the realm of producing systems, Wikipedia tells us that the term methodology is used to describe “a codified set of practices (sometimes accompanied by training materials, formal educational programs, worksheets, and diagramming tools) that may be repeatably carried out to produce software.” The methodology term is however not without problems, even though most researchers tend to favour it. However, some researchers dislike the use of the term “methodology”. Stamper (1988) for instance, uses the term methodology, but under protest. The only reason for Stamper, is to adapt to customary usage, but as he states “It would be better, as in philosophy of science, to speak of ‘methods’ when referring to specific ways of approaching and solving problems, and to reserve ‘methodology’ for comparative and critical study of methods in general; otherwise this vital field of study is nameless”.

Brinkkemper also finds the use of the term methodology incorrect. He finds the misuse of the term “methodology” instead of “method” to be a sign of immaturity of our field and that the use of the term “methodology”, in this meaning, should be abandoned. Rather, the term “methodology” should be restricted to scientific theory building about methodical information systems development (Brinkkemper, 1996). Thus, “methods” are the objects of study and “methodology” is the scientific field in which methods are studied. In this dissertation, I follow Stamper’s and Brinkkemper’s line of thought and refer the objects of study as systems development methods instead of methodologies.
Then, what is a method? Etymologically, the word comes from the Greek word *methodos*, which in turn is a combination of the two words *meta* (beyond, after) and *hodos* (way, journey). The etymological reference for the word could be in the vicinity of “A way to come beyond the point you are now” or something similar. Of course, the concept of method in this dissertation will be treated with a little more scrutiny. In order to be successful in the task of producing a workable and useful definition of method rationale, we must first start with some kind of analysis of the method concept itself. It will be necessary to review different conceptualisations of the method concept to create a valid starting point for our endeavour. Subsequently, this chapter will contain a review of the fundamental concepts in systems development methods as expressed by various researchers in the fields of ISD and ME. The chapter starts with an analysis of the field of ISD and how the concept of method has been used there. An analysis of the field of ME will follow and the chapter will end with a short conclusion concerning strengths and weaknesses between the two fields with regards to how they have treated the method concept.

### 3.1 Conceptualisations According to ISD

If conceptualisations of systems development methods in ME are used for formalisation, in ISD they are used to describe a more intangible phenomenon. Most definitions of what a systems development method is in the field if ISD are more centred towards verbal or written descriptions of the object of study. In essence, it is possible to state that ISD research emphasise relevance rather that rigour.

The various definitions range from somewhat vague ones, comprising general statements about what a method really is such as; a method is a way to "put specific concepts of rationality into practice" (Klein and Hirschheim, 1991), or more elaborated ones, saying that a method can be regarded as; "a full set of concepts and models that are internally self-consistent" (Henderson-Sellers, 1995). It really depends on which aspects of governance in systems development projects that the definer wishes to highlight. Cockburn (2000) also includes cultural aspects of the development team in the method term; “I use this term to denote everything about how a group repeatedly produces and delivers systems: Whom they hire and why, what people expect from co-workers. The processes they follow, their conventions, work products and even seating arrangements”.

Attempts have been made to formally describe what a systems development method is in the field of ISD as well. Röstlinger and Goldkuhl (1994) present a component based view on systems development methods, with an intention of combining flexibility with stability. The basic idea with method components is to use a rather high level of abstraction and structure the component’s content based on internal dependencies.

A component is always constituted of concepts (what to talk about), procedure (what questions to ask), and notations (how to express the answers)
(Goldkuhl, 1991). The concepts are the result of the underlying perspective of the method creator and direct the attention towards certain phenomena in a development context. The perspective also influences co-operation forms in which the method components are enacted and the frameworks in which method components can be integrated (Goldkuhl, 1994).

According to Röstlinger and Goldkuhl (1994) the requirements for classifying a method module to be a method component, is whether or not it represents a coherent picture for handling a specific and delimited problem in a systems development project. In comparison with the method fragment concept and the method chunk concept the method component has taken a standpoint when it comes to the question of granularity of method modules.

Hence, a systems development method can be regarded as a container of several method components.

![Figure 6. The method component concept (Goldkuhl et al., 1998)](image)

### 3.1.1 The Perspective of Systems Development Methods

Methods as such are conceived for reasons. This prescriptive dimension is called a perspective and it is a manifestation of a theory of how a systems development projects should be performed (Nurminen, 1988; Goldkuhl, 1991). According to Ågerfalk and Åhlgren (1999) a methods perspective is related to the method constructor’s principles, values, conceptions, experiences, categories, and definitions. The perspective can be understood as a philosophical paradigm and ultimately governs what is implemented in a systems development method, and what is not (Jayaratna, 1994).
As described in Figure 7 the perspective contains values and goals which
inflct on the way we understand phenomena in the real world. Based on our
values we categorise and define the world in order to interpret the various
phenomena we encounter. These elements can be regarded as goals and
ultimately be realised in a method description. Thus, a method description is
prescriptive conceptualisation, focusing our attention towards goals, values,
categories, definitions, and away from other phenomena in the real world.
Subsequently, all methods should be treated with some scrutiny and criticised.
The most fundamental aspect of this critique ought to be questions on what the
perspective includes, what it excludes and how this is manifested in the method
description.

Obviously, the point of being explicit about a methods underlying perspective
should be clear to anyone. It would help a lot if a potential method user could
ascertain whether a certain systems development method is a good choice just
by looking at the methods’ perspective. However, these aspects are often not
explicitly stated by the method creators. Very often, a systems development
method is so much more than just what’s in a book; “A methodology is a
recommended collection of phases, procedures, rules, techniques, tools,
documentation, management, and training to used to develop a system; we also
note the importance of the philosophy behind the methodology, or the set of
beliefs and assumptions underpinning it, explaining why the methodology
functions as it does. It may embody a belief that the key to successful
development is user involvement and that the users have a right to participate in
developments that affect their lives.” (Avison and Fitzgerald, 2003)

![Figure 7. A method in relation to its perspective (Goldkuhl, 1994)](image-url)
Some of the perspectives in systems development methods have been explicitly externalised and are thus more commonly known. Object orientation is a prominent example. The ideas put forward through the perspective concept in the field of ISD do not exist in the field of ME.

### 3.1.2 Methods vs. Methods-in-Action

A formalised systems development method described in a book is something quite different than a method actually being used in a development project. These methods can be developed in-house or it can be a commercially available method. There is also the possibility that the systems development method is a result from an ME project. However, ME research is not interested in studying how the method is being used in actual projects and how the method serves as a guideline for human action. In some way, ME researchers must take for granted that the users of systems development methods use them as they are described. The ISD research often disagrees with this picture. Here, the formalised method really becomes important when it is described and understood in use.

Described in a book, the formalised method is actually an ideal type and does not really exist in reality (Avison et al., 1998). The formalised ideal type only exists on a purely tentative level and as soon as the method is being used in reality, it ceases to be an ideal type and becomes situational, or a method-in-action (Fitzgerald et al., 2002). Research has shown that methods rarely are used the way the creators intended them to be used (Fitzgerald, 1997; Avison et al., 1998; Parnas and Clements, 1986). Rather, the method user enacts or instantiates a formalised method in action. In this instance the original formalised method does not longer guide the development process in its entirety, it merely acts as a template. Therefore, there is a distinctive difference between a method formalised in a book and a method enacted in-action.

The distinction between formalised methods and methods-in-action can be compared to espoused theory and theory-in-use (Argyris and Schön, 1995). Thus, using a method means that an ideal typical systems development method is interpreted and put into action, much in the same way a research method is interpreted and put into action in any research (Avison et al., 1998).

### 3.1.3 Frameworks and Models in ISD

ME research is, as already stated, typically devoted to structuring of method fragments or method chunks into complete systems development methods. In ISD these issues are covered by the use of frameworks or models to organise methods or method parts to be used through a development project (Andersen, 1994). These are outlines of development processes, generally divided in to some form of phases, each focusing a specific problem area. Each phase is the result of a judgement of the model constructor where the phase is defined and delimited as a part of the real world worthy of investigation (Avison and Fitzgerald, 1995). Examples of models are the commonly known Waterfall model which originates in the early 70-ies and the Spiral model (Boehm, 1988).
In more general terms, the term model is also used to denote resulting products in a systems development project. A Use Case model (Rumbaugh et al., 1999) for instance. This type of model also delimits a specific part of reality worthy of investigation but it does not focus on an outline of a development project. To distinguish between these two types of models, I will use the term development model to denote the models with this former type of ability or focus. Nilsson (1999) identifies two different ways to connect methods or method parts into development models or frameworks. Firstly, there is a need to integrate the methods or method parts in a way that ensures that useful results can be used later on in the development process. Nilsson calls this way of method integration a method chain where the chain metaphor is used to illustrate an inter-process integration occurring across phases in the development model.

<table>
<thead>
<tr>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP¹</td>
<td>MP²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method Chain</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 8.** Method chains and alliances explained with the Rational Unified Process phase model. Adapted from Kruchten (1999; Brinkkemper et al., 1999)

Secondly, integration can also be performed within a specific phase in order to increase the focus of the phase’s problem area. This type of integration is metaphorically called an alliance and illustrates an intra-process integration between methods or method parts within a specific phase of a development model.

In Figure 8, chains and alliances are illustrated in the development phase model of the Rational Unified Process (Kruchten, 1999). A chain or an inter-process integration is achieved through collaboration between method parts 1 and 2 (MP¹, MP²). This means that MP² draws on the results of MP¹. An example of this could be how a Business Use Case Model is used as input for the task of eliciting a Use Case Model.

The alliance or intra-process integration is illustrated through collaboration between method parts MP² and MP³. In the alliance, two method parts are used to enhance a particular focus within a given phase. An example of an alliance could be how the Use Case Model is used to refine the Software Requirements Specification in the Elaboration phase.
3.2 Conceptualisations According to ME

The use of the concept of systems development methods in the field of ME is primarily oriented towards how to describe systems development methods and parts. If ISD research emphasise relevance, ME tends to focus on producing rigorous solutions for how to find appropriate systems development methods for unique projects and organisations. The ambition is to handle and administrate the planning and engineering of new or tailored systems development methods for unique situational projects (Kumar and Welke, 1992; Brinkkemper, 1996). Therefore, the field of ME mostly treats the method as something that can be described and communicated. Fundamental for this task, is the question of how the method should be modularised, i.e. divided into useful parts. In the field of ME these modules are referred to as method fragments (Brinkkemper, 1996) or method chunks (Rolland et al., 1999).

Brinkkemper et al. (1999) propose that method fragments are classified according to the three dimensions: perspective, abstraction level, and level of granularity.

![Diagram showing the classification of method fragments into perspective, abstraction level, and level of granularity.](image)

**Figure 9.** A visual perception of the method fragment concept. Derived from (Brinkkemper et al., 1999)

The perspective dimension considers the product perspective and the process perspective. A product fragment is a deliverable, such as a document or a system module. A process fragment represents the activities to be carried out to produce the products.

The abstraction dimension consists of the conceptual level and the technical level. Method fragments on the conceptual level are descriptions of systems development methods or parts thereof. Technical method fragments are executable tools implementing methods (i.e., CASE-tools).

The granularity dimension concerns what level of granularity (or aggregation) a particular method fragment resides on. There are five possible levels: method, stage, model, diagram, and concept. A complete method, for example OMT (Rumbaugh et al., 1991), resides on the method level. A fragment at the stage
level addresses a segment of the development lifecycle for example *Detailed design* of OMT. A fragment on the model level addresses a particular view of the system, such as a data model or user interface model. A fragment at the diagram level concerns a specific view of a model fragment, for example, a *Class Diagram* showing parts of a *Class Model*. Fragments on the concept level are the atomic building blocks of the method, its process and notation. An example would be a modelling primitive such as *object*.

The fragments are used to formally describe situational methods that explain how project teams should work. The reason for describing methods in a certain way is connected to how the results of the research are intended to be used. One of the most prominent examples of method modularisation can be found in the field of situational method engineering (Harmsen, 1997). This variant of method engineering takes a radical starting point and argues for the need to construct new unique methods for each development project. This is done by modularisation of systems development methods into method fragments which are later stored in a method base. By the use of a specifically developed Method Engineering Language (MEL) the fragments in the method base can be administrated and manipulated. A unique situational method is then constructed from the method fragments stored in the method base (Harmsen, 1997).

In an attempt to tighten the coupling between process fragments and product fragments the notion of method chunks has been proposed. In the chunk concept these two types of fragments are integrated in one single module (Rolland *et al.*, 1999). In this way any systems development method can be regarded as a set of method chunks on different levels of granularity, thus a whole method can be regarded as a chunk in the same way a method fragment can constitute a whole systems development method.

Chunks are defined from a process perspective. This means that, in comparison with the method fragment concept, the core of each chunk is a guideline to a specific process and its related product fragments (Ralytė and Rolland, 2001). This guideline embodies the knowledge needed to achieving an intention in a development project. Hence, the guideline has an interface, which describes when it can be used and a body providing guidance to achieve the intention. An intention in this context means to construct something useful in a development project. Example of an intention can be; *Elicit a Use Case*. Thus an intention can be regarded as focusing on tasks rather than the actual goals being achieved through that task. Between the chunks there are interfaces describing which situations need which intentions and the input needed (Ralytė, 2002). The chunks are then used to construct new methods in a similar way method fragments are. Other conceptualisations within ME research have been proposed including Firesmith (1998), Henderson-Sellers (2007). However, these conceptualisations are very similar to the conceptualisations already presented and therefore, will not be further elaborated upon.
3.3 Strengths and Weaknesses in Existing Method Conceptualisations

Comparing the conceptualisations of systems development methods in ME and ISD will show both strengths and weaknesses in the two fields. These strengths and weaknesses present challenges for how the systems development method concept could be perceived in EME. Problems of communicating research results in a way that can be perceived as meaningful from one field of research to the other can be connected to how the two fields define and use the concept of systems development methods. For instance, statements about the perspective dimension is more or less impossible to use in the field of ME since these aspects of systems development methods are not covered in their definitions of what a systems development method is. Another example of communication problems can be found in a situation where a proponent of ME tries to formally describe how a systems development method should be used to a person from the field of ISD and not differentiating between the systems development method itself and the systems development method-in-action. The reasons for this are found in the fact that ME does not have a method use or action level. They simply calculate that the engineered method will be used as intended, although a lot of research indicates that this seldom is the case (Fitzgerald, 1997; Parnas and Clements, 1986).

The field of ME is perhaps too formalised. At least, there seems to be an ambition to formalise conveyable knowledge of systems development methods. The problems the field of ME is facing are concerned with demarcations of what actually can be formalised and conveyed. One apparent problem with the method fragment concept is that the formalised fragments lack congruence, especially in the granularity dimension. In this dimension a fragment can represent a whole method as well as just one single concept in that method. Needless to say, this can cause problems, at least in situations where methods are communicated. On the other hand, the relatively high degree of formalisation makes it possible to develop method management systems such as the ones proposed in Harmsen (1997) or Ralyté et al (2003). This also calls for principles describing how the systems development methods could be communicated. In the field of ME, the formalised descriptions are used as the common ground for communication about systems development methods, i.e. an method engineer is supposed to understand what a method fragment is when it is presented to him/her.

In comparison with the field of ISD, the perspective dimension seems to be missing in ME. This is interesting since these issues are partly covered in Brinkkemper’s (1996) definition of what a systems development method is “A method is an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products”. An interpretation of this definition could put the perspective dimension from the field of ISD in the phrase concerning “way of thinking”, but
then again, we can not be sure Brinkkemper means it that way. When it comes to formalisation of the systems development method, any attempt to formalise goals or values connected to a systems development methods perspective is missing. The notion of intentions in the method chunk concept does not translate in to the perspective dimension of ISD since an intention is a description of what needs to be done rather than the actual reasons why something needs to be done.

The modularisation efforts in the field of ME could be useful in situations such as method communication, and use. In communication situations, method modules serve as tools for conveying method knowledge in appropriate portions. Modularisation in method usage situations serves as checkpoints and clarifications of how project results can be used. In these cases a certain degree of formalisation could be beneficial since they can serve to clarify issues concerning systems development methods. These aspects of method formalisation have been overlooked by the field of ISD. Figure 10 summarises how the two fields view systems development methods and also point to strengths and weaknesses in the two fields.

Figure 10 Strengths and weaknesses in ISD and ME

The figure indicates that the aspects considered to be strong in one field are considered weak in the other and vice versa. One way to handle the identified problems would be to define a new conceptualisation of what a systems development method is. Evaluation criteria for this new conceptualisation would of course be related to the discrepancies identified in Figure 10. These criteria would capture the individual strengths of the conceptualisations in the two fields and enable better possibilities to communicate systems development knowledge between the two fields.

An important task that will be undertaken in this dissertation is the task of formulating such a conceptualisation and show how this conceptualisation can
be utilised for EME and communication of systems development method knowledge in systems development method research.
4. Method Rationale

Going back to chapter 1, providing a synthesis between the fields of ISD and ME through EME is one of the main points of this dissertation. One major step in this endeavour is the definition of a concept of method rationale that can act as a tool for communication and a solution space between the two fields of ME and ISD. Thus giving ME a chance to break new ground and find new areas of interests. This is done through formulation of a synthesis in the form of a conceptual bridge that explicates what method rationale is and primarily how it used in the two fields of research. This chapter focuses how the knowledge inherent in systems development methods is treated in papers on method knowledge in both ME and ISD.

The chapter will be devoted to a synthesis of how method rationale could be treated, enabling the concept of method rationale to play an equally important role in both fields and creating possibilities to create a common ground of understanding of the concept. Naturally, my ambition is also to define the concept in a way that can be understood as meaningful and useful in all types of systems development method research. Basically, the definition will have a foundation in how the concept is used and perceived as meaningful in the two fields of ME and ISD respectively.

Hopefully, proponents of ME can benefit from relating their concept of method rationale to the theories on the roles methods play in social action. On the other hand, the field of ISD can benefit from studying how ME research generalises its results. A synthesis of this kind would help facilitate the possibilities to produce rigorous systems development methods without compromising with the relevance.

This is achieved through a literature review on the topic and formulation of a synthesis capturing the phenomenon in a way that can be perceived as useful from both perspectives. I will start off in a general discussion on rationality and what can be defined as rational action before going forward and actually describing how these concepts have been treated in the two fields of research.

4.1 Chapter Overview

This chapter is the first in a series where I start to present my findings. The following chapters will also have a chapter overview section describing the chapters’ role in relation to the model describing the entire research process in Figure 4. Each chapter overview section will present a part from Figure 4 and explain how it has been treated in the following chapter. The overviews will also be in the form of UML Activity diagrams (Booch et al., 1999) just like Figure 4.

This chapter is the result of a conducted literature review, following the basic concept centric approach as suggested by Webster and Watson (2002). The details of how this approach was applied are described in chapter 2.
The chapter includes one activity *Literature review*. The primary inputs for this activity are my research question and an array of different research papers. The papers are analysed and conclusions are drawn about the concept of method rationale. Thus, an initial definition of method rationale is considered to be output from this chapter. My research question also addresses issues concerning representation of method rationale. This involves questions of how a systems development method could be represented in order to be able to communicate its method rationale content. In the end of this chapter I present, and argue for the need for, a systems development method module notion as a way to overcome the identified problems.

A model describing the details for this chapter is presented in Figure 11 below.

![Figure 11 Chapter 4 details](image)

### 4.1 The Concept of Rationality

In order to discern what method rationale really is, it is important to first take a look at the concept of rationality itself. Nonetheless, since it is an important criterion when judging whether or not certain statements on method rationale really represent true rational behaviour.

One way to approach the concept of rationality is through Weber and his fourfold typology of social action (Weber, 1978) This typology is oriented towards explanations and intentions behind human behaviour. The typology is easily understood as hierarchical even though it must not necessarily be interpreted that way:

*Traditional action*, which typically is defined as behaviour governed by habits. We can understand this type of action in a development situation, as a state where there might not exist a method, or that there is a lack of understanding concerning the underlying rationale in the method. The developers basically do their thing as they are used to.

*Affectual action* is based on the actors’ current emotions. An example of how this type of action can be found in a development situation is a situation where the developer does not want to do certain steps because s/he finds them boring. According to Weber, these two types of action do not represent true rationality. In fact they don’t even represent irrational actions. Weber rather finds these two types of action to be non-rational as opposed to irrational. The difference
between these types of actions is that irrational actions are founded in some type of reasoning, whilst the non-rational actions are not.

Value rational actions are founded in a set of values of some sort. This set of values takes past, present and potential future into consideration. The rationality in a certain action lies in the ambition to strive towards and fulfil certain value bases.

Means-end (goal) rational actions can be understood as the focusing of an action towards specific goals. According to Habermas et al. (1996) and Schluchter (1981) the means-end rational action involves rational consideration of alternative means to the end, relations of the end to secondary consequences and the relative importance of different ends. Ultimately this means that means-end rational actions take different side effects into consideration. This situation is related to a mature use of systems development methods where the user understand the underlying purposes of the method and how they are related to its proposed actions. Maturity is an important concept in this dissertation since I hope that the explication of method rationale could help users of systems development methods to become mature more quickly. To support developers in gaining experience and speeding up the process of becoming mature professionals the concept of method rationale must be understood in a goal-rational context. Therefore, Weber’s goal rational action can serve as critique against how method rationale is used in the two fields but more importantly as an argument for a synthesis in the form of EME.

4.2 The Concept of Method Rationale

The concept of Method Rationale has previously been defined as “information regarding decisions that leads to a specific meta model” (Rossi et al., 2000). This definition originally comes from the field of method engineering and has its foundations in the concept of design rationale (Conklin and Begeman, 1988; Dix et al., 1998). The basic advantages of using design rationale according to Dix et al (1998) are:

- Communication mechanism among design team to communicate past critical decisions, what alternatives were investigated, and the reason for the chosen alternative.
- Transfer of design knowledge between projects with similar rationales
- Encourages deliberation and explicit consideration of alternatives.

Basically, design rationale is focused towards possibilities to capture and communicate decisions concerning a design. notion has been transferred into an ME context with similar meanings. Work of this kind can be seen in (Oinas-Kukkonen, 1996; Rossi et al., 2000; Rossi et al., 2004). In the field of ISD the concept of method rationale is more oriented towards inherent qualities of the method (Klein and Hirschheim, 1991; Goldkuhl, 1994; Iivari et al., 2000), method knowledge in action or use (Fitzgerald, 1998a; Parnas and Clements,
It is also possible to find similar ideas in other research (Russo and Stolterman, 1997; Ågerfalk and Åhlgren, 1999). The concepts used in these papers differ slightly but the basic notion is similar, i.e. the concept of intentions is frequently occurring. All in all, no one has tried to formulate any homogenous theory on method rationale.

### 4.2.1 Method Rationale in ME Research

By meta model (Rossi et al., 2000) mean a specific method that can be used in different development situations. The main point they present is that methods in use are subjects to change. The persons using the method make decisions during method use and these decisions shape or adapt the method in an organisational context. Another way to put it would be to say that method use, leads to method adaptation. Rossi et al. (2000) argues that development organisations should try to capture the arguments behind these adaptation decisions. The idea is that this could facilitate the task of assessing the current development process, as well as introducing new co-workers to the organisation specific way of working. Their basic idea is to collect and explicate tacit knowledge (Polanyi, 1983) about adaptations of methods, mainly for in-house use.

Rossi et al. discuss two different types of method rationale: method construction rationale and method use rationale. Rossi et al. (2000) focus on method use rationale and how this type of method rationale can be used to explicate otherwise tacit knowledge about method adaptations to co-workers. The concept of method construction rationale is not thoroughly explored in their paper.

To understand the concept of method construction rationale better we need to look at the area of research known as Situational Method Engineering. In this field of research there is a lot of focus on the underlying arguments behind decisions that lead to a specific situational method, i.e. a method that has been constructed to meet the requirements raised by unique development situations. A lot of work has been done in this area (Harmsen, 1997; Brinkkemper et al., 1999; Ralyté et al., 2003) and the common denominator seems to be the ambition to create situations where the development team can use a uniquely constructed method that meets their demands in every aspect.

Figure 12 describes how method rationale has been focused and perceived in the field of ME. The figure is divided into three hierarchical spheres which describe boundaries between identified concepts and the primary focus of method rationale.

The highest sphere is the Method engineering sphere. In principle, most of the research in the ME field falls into this sphere. The core concept in this sphere is the Meta model. A meta model describes a systems development method on a higher level. It is a description of a method rather than an enactment or instantiation. The meta models are usually managed in some way and a lot of
effort has been put into management of meta models (Brinkkemper et al., 1999; Harmsen, 1997; Karlsson, 2005). According to Rossi et al. (2004) the meta models can be found rational by focusing on the underlying decisions that lead up to its design. These arguments are expressed as the systems development methods’ Method construction rationale which in turn is a collection of Design decisions concerning the appearance and design of the systems development method. Basically, as described above, a lot of ME research is oriented towards the production of Situational methods which are either new designs or redesigns of systems development methods (Brinkkemper, 1996; Kumar and Welke, 1992; Ralyté and Rolland, 2001). This task is most commonly carried out by gathering some kind of Requirements which is raised by a particular Development Situation (van Slooten and Hodes, 1996). The requirements form the arguments for the design or engineering of the situational method which then is applied in the development situation.

**Figure 12.** Method Rationale in ME. Unbalanced focus lacking action dimension

Right below the method engineering sphere we find the Method use sphere. This sphere is where the actual use of the systems development method takes place. Not much of this is covered by ME research, hence the fuzzy cloud representing the development situation.

The third and final sphere is the Method reflection sphere. Here reflection and experience concerning why a certain systems development method was redesigned during use is captured and formalised. According to Rossi et al. (2004) this can be done through use of computerised tools capturing and expressing the collection of arguments leading up to the redesigned method. This collection is to be understood as the Method use rationale. In essence, redesign or redefinition of a method means that the knowledge concerning the
underlying meta model is rethought and altered. Figure 12 expresses this consequence as the method use rationale is said to present arguments for redesign.

From a Weberian standpoint, the field of ME fairly well captures the possibilities to act means-end or goal rational, at least in the method engineering sphere and the method reflection sphere. What ME research miss is the actual usage of systems development methods in actual projects. In their solution oriented fashion they conclude that a situational method is more or less actually used as intended, otherwise their solutions would be useless. A problem with this position is that there is very strong empirical evidence that users of systems development methods rarely use the methods as intended (Parnas and Clements, 1986; Fitzgerald, 1997; Introna and Whitley, 1997). This problem results in a model describing ME research as having an unbalanced focus of method rationale, totally lacking an action dimension when it comes to actual method use, and only allowing feedback through the use of computerised tools capturing method use rationale.

4.2.2 Method Rationale in ISD Research

In the field of ISD, it is more common to speak of the concept of method rationale by connecting the concept to social action. Basically a systems development method is considered to be a way to put specific concepts of rationality into practise. They are mental constructs, structuring complex sequences of social action and giving them meaning (Klein and Hirschheim, 1991).

A method is basically a way of structuring thinking and actions. They are to be considered as reflecting particular perspectives of reality based on a set of philosophical paradigms. These paradigms contain the notion of “why” a certain step is prescribed (Jayaratna et al., 1999). This is reflected in Figure 13 describing how the concept of method rationale is focused in the field of ISD. Like the field of ME it also has three spheres.

The first sphere is the Method conception sphere. I use the term conception instead of construction since ISD research emphasises that a lot of systems development methods have a foundation in best practices rather than in an engineering process (Kruchten, 1999; Goldkuhl, 1994; Beck, 2000). In either case, the proponents of ISD research sometimes see the systems development method as an expression of some form of public rationality (Klein and Hirschheim, 1991; Russo and Stolterman, 1997). This public rationality encompasses the underlying arguments as to why the method has a certain appearance. An important part of this rationality is the Philosophical paradigm which delimits the systems development methods’ perspective on reality (Jayaratna et al., 1999). This is done through a set of concepts founded in Goals and Values. Aligning with emotive theory (e.g., Ayer, 2001), a value can be understood as an ethical judgement that is not an assertion or report, but is an
expression of feeling or attitude. That is, values are imperatives that cannot be judged as true or false.

Figure 13. Method Rationale in ISD, Equal focus, poor feedback

The goals and underlying values motivates why an Ideal typical method has a certain appearances (Goldkuhl, 1994). The importance of “why” is also stressed by (Yu and Mylopoulos, 1994) and the total of the method conception sphere aims to give enough arguments to explain this issue.

The next sphere is the Method-in-action sphere. This sphere aims to explain how the systems development method is used in action. The concept of method-in-action appreciates that systems development methods exist in enactment. The difference between a systems development method at the method-in-action sphere and a systems development at the method conception sphere can be likened to Argyris and Schöns’ (1995) concepts of espoused theory and theory-in-use.

In this sphere, the ideal typical method is used as a method-in-action. As such it supports Social Action carried out by various Actors. This usage situation does not only involve usage. It also involves the adaptation of the method-in-action.

The third sphere is a Method reflection sphere in the same sense as in ME. Here lies the private rationality of the actors involved in the systems development project (Russo and Stolterman, 1997). Residing here are the preconceptions that influence the actor during his/her enactment of the method-
in-action. These are attitudes such as Human bias or Intentions. I have narrowed it down to these two since they represent two various modes of attitude. Human bias is essentially non-rational or irrational whilst intentions essentially are better formulated and hence, rational. Thus, the private rationality of actors must not necessarily be considered as fully rational altogether.

The private rationality also contains elements of knowledge. These elements, Skill and Experience, are often in the form of tacit knowledge (Polanyi, 1983). The total of the private rationality is what defines the actor as a person. It is his/her preconceptions and frame of reference in the world that influence how s/he acts in real life. According to a hermeneutic ideal the preconception does not only govern the actions taken. It is also affected by the actions taken as these create new preconceptions in accordance with the hermeneutic spiral (Marc-Wogau, 1993). Personal knowledge about systems development methods in general and methods-in-action in particular is thus considered to be an important personal development process, building on a cumulative knowledge base.

Even though some of these knowledge elements could benefit from articulation along with the attitudes, the researchers in the field of ISD rarely try to explicate the private rationality into generalisable knowledge which possible could affect the numerous ideal typical methods that exist. This problem is illustrated in Figure 13 through the use of dotted arrow going from the private rationality in the method reflection sphere to the ideal typical method in the method conception sphere. Consequently, the field of ISD can be considered as missing the important aspect of feedback into generalisable knowledge.

From a Weberian standpoint the research of ISD focus the concept of method rationale equally through all three spheres. The main critique concerns the lack of feedback other than the interplay between the actor and the private rationality.

4.3 Synthesising an Ontological Framework for Method Rationale

The two fields have strengths and weaknesses. The field of ME has an unbalanced focus and lacks an action dimension. On the other side, the field provides rigorous, tool supported approaches to method tailoring and management of method use rationale. The field of ISD has an equal focus but has poor feedback. On the other side, the field provides for a relevant appreciation of how a method can serve as support in action. It also captures the philosophical dimension of methods and how actors develop a deeper understanding of systems development practise. This identified discrepancy between the two fields is a starting point for a formulation of a synthesis in the form of EME.

To complete this task, we must define method rationale ontologically. There is a need to chisel out and define the conceptual constructs that can act as powerful communication tools between and in the spheres. These constructs should be able to express both method construction rationale, as well as method use rationale. Russo and Stolterman (2000) state that systems development
methods only are of value as long as the intentions behind their design are made explicit and understood by all parties. I would like to add ‘in all three spheres’.

4.3.1 Method Rationale in Extended Method Engineering

In my synthesised model describing EME shown in Figure 14, the three spheres is a combination of the two fields of research. Since the field of ME is naturally strongest in the method engineering sphere the top sphere will keep this name. The second sphere must bridge the problem of ME not being able to focus systems development methods action dimension. In ISD this is covered by the notion of method-in-action. Subsequently, since ISD is very strong in explaining how systems development methods are actually used, the second sphere will be called the method-in-action sphere. The third method reflection sphere is essentially the same in both original fields, but with a difference in focus. In ME the reflection basically concerns method use rationale whilst ISD research concerns the actors’ private rationality.

In the different spheres, the original elements have been combined, renamed and synthesised into new elements describing how the concept of method rationale could be focused and used in both fields of research. The synthesis aims to provide for possibilities to overcome the weaknesses of the two fields by giving the rigorous field of ME an opportunity to draw on the relevance provided by the field of ISD. At the same time, the synthesis also aims to overcome weaknesses by providing rigorous tools and approaches to the relevant field of ISD, giving them better possibilities to generalise their results and communicate the tacit dimension of systems development method knowledge.

The method engineering sphere has been reshaped to reflect what method construction rationale is. In reality, very little differ between the notion of method construction rationale and the notion of public rationale. The result is a combination into a public method construction rationale. Both aspects are equally important for pinpointing what constitutes a method, separated from any usage situations. This new category contains the philosophical paradigms covered in Figure 14 as well as the underlying goals and values. Together, they explain why certain design decisions concerning the ideal typical meta model have been made.

The process of gathering requirements for adaptation of ideal typical meta models into situational methods is also covered in the method engineering sphere. An important difference is that the input to the requirements does not only come from the development situation, but also from the actors’ private rationality.
The second sphere is, of course, the method-in-action sphere. This sphere basically defines the fuzzy development situation from ME with the help of ISD terminology. It also allows for adaptation of systems development methods through use; something considered ad-hoc by the field of ME, but still representing a widespread behaviour according to ISD studies. Using the method in the method-in-action sphere means that the actors realises the goals and the underlying values by enactment of the method in various development situations. It also means that the method user is influenced by his/her private rationality which resides in the method reflection sphere as goals and values are used in a hermeneutic process of interplay between the method-in-action sphere and the method reflection sphere.

The third sphere is still regarded as the method reflection sphere but the reflection is divided into two parts, one part that can be communicated easily and one part that is more difficult to fully express. This is done by appreciation of the Private rationality dimension as expressed by the field of ISD with one important clarification. In essence, intentions are reasons for actions and are thus goal oriented. Goals do not exist by themselves. The describe target states which
someone has deemed valuable for some reason. Thus, the intention element is divided into and defined as a combination of goals and values.

Outside the private rationality we find the method use rationale. In ME this was defined as captured arguments from method usage situations. According to ISD, method use rationale should rather be connected to and be considered a deliverable from the private rationality since the private rationality captures any reflections or experience from method usage situations already when it happens, long before any computerised tool can. Thus, the method use rationale is tighter coupled to human aspects than before. This method use rationale can then be used as arguments for redesign according to traditional ME. As a consequence this model provides for the possibilities of achieving double feedback. It does not only capture the feedback interplay between the actor and his/her private rationality. It also covers the feedback from method use situations into arguments for redesign of ideal typical methods. From a Weberian standpoint, the model focuses goal rational thinking equally through the three spheres.

4.3 Defining Method Rationale

So far I have shown how the concept of method rationale has been treated in the two fields of ME and ISD. Left to do is to define the concept in a way that can be perceived as meaningful in both fields of research. This is absolutely necessary if future EME research will have an opportunity to explore new areas and hopefully conduct research of a more general systems development method orientation. The conceptualisation would also facilitate knowledge transfer between researchers typically belonging to either field and would be able to serve as a common ground for developing new types of research on systems development methods, research that will be able to transcend boundaries between fields that previously have been regarded as incommensurable. The notion of method rationale as being constituted by goals and values yields a model of method rationale as being shown in Figure 15.

In line with this reasoning, I propose two different kinds of sub-rationale forming method rationale: (1) method anchored in goals, referred to as goal rationale, and (2) goals anchored in values, consequently referred to as value rationale.

The goal rationale association expresses that the method has a goal related to its proposed activities. Everything in a systems development method is prescribed for a reason (Klein and Hirschheim, 1991). The goal rationale relation expresses that the proposed activities are intended to fulfil the goals the method creator finds to be important to fulfil. Goals on this level are typically very general. The RUP states the goal of is “… to produce, within a predictable schedule and budget, high-quality software that meets the needs of its end users” (Kruchten, 1999). The goals do not really say how you are supposed to be able to do just that, but they claim that you will reach a certain target state by following the suggested method.
Goals can be regarded as desired target states. This means that certain target states are considered valuable to attain. The value rationale association represent the relation the target states, or goals, have to their underlying values. Examples of values in the RUP are “customers deserve high-quality software” or “customers deserve a system that meets their needs”. Hence, there is a value based judgement behind the reasoning that motivates why it is desirable to reach every one of the target states, or goals, the method tries to reach.

Furthermore, goals can be related to each other in goal hierarchies. An example of this is when a goal is viewed as a means to achieve another (higher) goal. Similarly, values can be anchored in other values. I define these two properties of rationale as goal achievement and value anchoring, respectively. In addition to goal achievement, there is a possibility that goals contradict rather than complement each other – hence there is an additional goal contradiction relation defined over the set of goals. Similarly there is a value contradiction relation defined over the set of values. In a well conceived systems development method you would not necessarily find any contradictions of this type. However, we must assume there are possibilities that contradictions of this type might exist within a systems development method.

As depicted in UML Class Diagram notation in Figure 15, the method is related to at least one goal, and each goal is related to at least one value. This means that any method is supposed to directly or indirectly contribute to the overall goal of using the method. It also means that every goal has a counterpart in at least one value. Analytically, this implies that the reason for using a particular method can be based on (a) its contribution to other higher-level goals, and (b) its realisation of (parts of) the systems development method’s underlying philosophy as expressed by identified values. Goals not belonging to either category (called intrinsic goals) are goals for their own sake and are...
questionable in a rational development process. However, one would not expect to find any intrinsic goals in an available systems development method such as, for instance, the RUP. However, in order to fully understand the concept of method rationale, we must assume that contradictions between goals or between values might exist. Therefore, we include these associations in our model over method rationale in Figure 15. If we did not do so, the model would be conceptually incorrect. This definition of method rationale can be used from either direction (from method engineering sphere and down, or from method reflection sphere and up) in my synthesised model in Figure 14, enabling double feedback through use of reflection about the method in relation to its goals and values in the method reflection sphere and the possibility to express private method use rationale through the goals it fulfils and values it achieves in a way that could be useful in the method engineering sphere. This is a key point in my synthesis since it enables possibilities to communicate method knowledge across all three spheres. From the method engineering sphere, goals and values convey method knowledge through method construction rationale into the method-in-action sphere. This method concept has a possibility to be used in the field of EME as it would bring a typical method engineering concept to be used in areas where they previously have not been applied.

4.3.1 Rationality Resonance

Using an externalised systems development method means that the method user is trying to achieve certain goals and considers the method to give adequate support for his/her efforts. The goals s/he wants to achieve are considered as useful in order to produce the system s/he wants to produce. As the systems development method also has goals, using the method, is an attempt to create a harmonisation between the goals the method is providing with the goals of the developer. On a very coarse level, an example of this could be that the developer is trying to produce a system that meets the customers' needs. If the developer then chooses to use a systems development method that claims to aim for that goal, there will be a possibility for harmonisation between the goals.

This kind of state is called rationality resonance (Russo and Stolterman, 1997) and describes a state where the public rationality of the method coincides with the private rationality of the actor in a usage situation. The actor can then use the method in a well informed fashion. This notion is similar to the argument for value congruence between systems development method and its user as being suggested by (Jayaratna et al., 1999).

Going back to the synthesised ontological framework for method rationale in Figure 14, the possibilities of actually reaching rationality resonance is limited to one sphere, the method-in-action sphere. However, since rationality resonance is a harmonisation of public rationale which comes from the method engineering sphere and private rationality which comes from the method reflection sphere, knowledge elements of the same type must come into the method-in-action
sphere from the other two spheres. If they are not of the same type, it would be impossible for them to harmonise for obvious reasons.

Figure 16 The developer’s intentions related to goals and values

Thus, the only elements possible to bring into the method-in-action sphere which also fulfil the necessary level of congruence are goals and values. In the method-in-action sphere, goals and values are not only manifestations of the public rationality expressed through a systems development methods’ philosophical paradigm, nor just manifestations of the private rationality of the actor. They are descriptions of how the public rationality provided by an externalised systems development method is used in harmony with the private rationality that comes from an individual user in the method-in-action sphere.

Figure 16 describes how the developer also has goals and values that form the basis of his/her intentions in the same way that a systems development method is a manifestation of the intentions of a method creator. This is the developer’s private rationality. As great part of the conceptual models correspond, it is natural to see rationality resonance can be achieved. It basically means that there is an additional association between the method, the developer and a goal. This would yield a conceptual class diagram as illustrated in Figure 17.

As depicted, rationality resonance occur when there is correspondence between the activities suggested by the method, the intentions of the developer and a particular common goal, a goal that is founded in a common value base. The reasons for choosing a multiplicity of * is that it is possible that rationality resonance might actually not occur, as well as it might occur several times during method-in-action.
An alternative understanding of what takes place during a state of rationality resonance could be explained through the model depicted below in Figure 18. In the model, rationality resonance is represented as an intersection between the two kinds of rationality. In relational algebra the expression Public Rationality ∩ Private Rationality describes this intersection between public rationality and the private rationality.

**Figure 17** Correspondence between method and developers intentions creating rationality resonance

**Figure 18** Rationality resonance

### 4.4 Requirements for a Method Module Notion Capable of Carrying Method Rationale

So far we have examined the concept of method rationale from a perspective that takes the entire systems development method into consideration. The private rationality of the individual developer has also been treated as something straightforward and self sustained. In reality, things are rarely this simple. A modern systems development method such as the RUP contains a multitude of activities, diagrams and concepts, all suggesting various actions in order to
execute, draw or find them. It is not likely that a state of rationality resonance truly can be reached through the entire development process. There will hopefully be moments where the method is used as a method-in-action with goals and values harmonising and a developer that is fully aware of what s/he is doing and why. At the same time it is likely that there will be moments when a developer is not really sure why s/he follows certain suggested steps. Maybe, s/he has lost sight of the ultimate goals of the system s/he is developing because s/he has been tangled up in some detailed specification for a long time. It could also be the case that s/he only does what s/he has been taught or that s/he follows the example of others without really knowing why. These situations might ultimately lead to a situation where developers choose not to use methods as they find them unnecessary. Empirical studies have shown that method use tends to decrease as developers get more experienced (Fitzgerald, 1997) and this could be a reason why.

In order to understand the role method rationale plays during the process of developing a system it is necessary to be able to track where and how in the process of systems development rationality resonance occurs. The how is still pretty straight-forward. It still comes to finding situations where the value base and goal rational behaviour of the developer is corresponding to that offered by the method being applied. The where is more difficult to imagine. Naturally, during the course of a systems development project a number of activities take place. If rationality resonance will be possible to identify, there must be a way to divide a systems development methods into manageable parts. These parts must be possible to regard as single units and they must be possible to regard as self contained. This means that everything that is important to know about a method must be possible to identify. For instance, the method modules must contain suggested steps and explanations for how the developer should execute them. This also involves descriptions of the concepts being used and possibly a description of the role that is responsible for executing the suggested steps. As the method modules are to be considered building blocks for an entire systems development method there must be a possibility to connect the modules into a complete method.

The analysis of method rationale in the previous sections shows that are differences in how systems development methods and method rationale are perceived in the two fields of research I have studied. It also shows that it is possible to combine knowledge from these fields into a synthesis that could provide a new approach to systems development methods and enable method rationale to play a central role in all spheres of systems development method use.

In achieve a viable synthesis it will be necessary to find a common ground for systems development methods. That means that it must be possible to talk about these phenomena in a way that would make sense in any method research setting or tradition. It also means that knowledge transfer, through method rationale
between the, before mentioned, method use spheres will be possible through EME. This will ensure that it will indeed be possibilities to perceive the systems development method phenomenon in a similar fashion all through the total of possible areas of method use.

The method use spheres cover all relevant areas of method use. The spheres cover the Method engineering sphere, which is useful for denoting any method construction or conception activities. Choosing the term ‘engineering’ to define the sphere is not an ambition to enforce the idea of an engineering approach to all research focusing on method construction or conception. It is rather a way to emphasise the need for a more stringent approach to method representation through the use of method rationale. The ME field has had long experience in this area and the sum of its knowledge on method representation through their use of method bases and method engineering languages (Brinkkemper et al., 1999) will provide for a useful stance on the need for recording conceived methods by the use of engineering principles rather than a need for engineering approaches when constructing them. It emphasises the need to explicate the method rationale and to record it in manageable manner. This need is to be interpreted as a requirement on the representation form which will function as a common ground for how systems development methods can be perceived. Only by finding this common ground for a form of method representation that can be perceived as meaningful by researches from the fields of ISD and ME it will be possible to let ME research extend its previous boundaries and ensure true knowledge transfer through method rationale between the spheres illustrated in Figure 14.

The method-in-action sphere requires a method conception that can be used for discussions concerning method use. As the field of ME totally ignored this sphere most input for the definition came from the field of ISD. The most significant difference is our proposal for a generic method concept with a foundation in the method engineering sphere functioning as a communicative tool through the sphere of method-in-action. This would also facilitate the process of conveying requirements raised by the unique development situations back to the method engineering sphere.

The method reflection sphere also raises certain requirements. The private rationality this sphere tries to capture must be possible to represent as transferable knowledge back to the method engineering sphere with the same basic principles as when knowledge transfer is occurring between the method engineering sphere and the method-in-action sphere. Similarly, this calls for a need to find a common form of representation of the systems development method that would be meaningful in a situation where various arguments for redesign were raised and needed to be communicated to someone in the method engineering sphere.

By ensuring that method rationale can receive equal focus through all spheres we will be able to argue for a synthesis between the fields of ME and ISD.
through EME. In the analysis above I have argued for a common understanding, or denominator, that can act as a foundation for this synthesis. The foundation is method rationale and a method module concept able to carry this knowledge dimension in a way that can be perceived as meaningful in the three defined fields of method research. To enable possibilities to have a method concept capable of truly carrying the method rationale dimension we must find a modularisation principle that can ensure an appropriate level of granularity. There is always a risk that the method module covers too much of its content. This could complicate the possibilities to internalise the method module and there would be a risk that its inherent rationale dimension would not be as present as it ought to be. Conversely, a method module that covers only fragments of a method could be misunderstood and difficult to internalise since it may not provide enough information to be perceived as meaningful. The risk would be that the method module in itself did not cover enough to carry any rationale dimension. This would yield a situation where a possible state of rationality resonance would be difficult or nearly impossible to identify. An example of this would be a module concept that only covered goals or a diagram specification. With this level of granularity the intended method user would not have access to any actions needed to realise the goal or the values realised by the goal. Another example could be a module that only covered actions. In this scenario, the method module does not present any rationale for why these actions are proposed. There are no defined goals of the actions and there are no goals presented for what the module intends to produce in terms of, for example, a document. Nor are any values fulfilled by taking the proposed actions presented. This could lead to a situation where the method module would run the risk of being interpreted out of context and could not function as a communicative tool between, for example, the method reflection sphere and the method engineering sphere. Hence, a modularisation principle with a too narrow perspective would not be able to use when studying instances of rationality resonance.

All in all this analysis results in requirements for how a method modularisation principle capable of carrying method rationale should be used in EME research to ensure fruitful knowledge transfer between the method use spheres and the two existing fields of research. The requirements can be summarised as follows:

- Possibilities to create equal theoretical and practical focus on method rationale in all method application spheres.
- Possibilities to express and describe all instantiations of method use.
- Possibilities to modularise a systems development method into consistent and coherent units, capable of giving feedback between the spheres and also serve as a common ground for EME.
PART III – Method Rationale and Method Components in Context

*Do not follow where the path may lead. Go instead where there is no path and leave a trail.*

*Ralph Waldo Emerson*
5. Method Rationale in Method Implementation

This chapter explores the role method rationale plays in a situation where a systems development method is adopted by a large development organisation. Method implementation is not just a phase where an organisation goes from totally informal development processes to formalised and name given industry standards such as RUP. Implementing RUP as a systems development method is not merely conforming to a particular standard in order to pass off as more quality aware than before. Rather, the organisation is trying to create a development organisation with certain characteristics. The characteristics can vary, but generally they address issues such as; repeatability, transparency, quality or the ability to manage and improve their development process. An implementation process is a cumbersome and difficult task. It involves many aspects such as creating awareness, assessing the organisation, creating motivation, setting goals and identifying risks (Fitzgerald et al., 2003; Bergström and Råberg, 2004).

The people responsible for the implementation phase must take the organisational know-how and existing practises into consideration. This means that they must analyse and understand the informal development processes that are already present in order to properly assess the target organisation and to understand how, why and where external method support would best be implemented. These types of analytical tasks require that the people involved are aware of the underlying method rationale of RUP, either explicitly or implicitly. The reason for this is that the communication of, and about, systems development methods have a foundation in this rationality as it is giving the proposed activities meaning and reason. A very straightforward way of putting it is to say that method implementation is difficult without the help of method rationale. The reason for this is because the knowledge dimension manifested with the help of method rationale is ultimately what is in the focus of attention. Implementing a method means that this knowledge dimension is communicated to an organisation and method rationale is what gives the knowledge dimension substance and reason.

This chapter reports on a method implementation case at Posten IT. It focuses on the role method rationale plays at the early stages of method implementation. The chapter aims to explore and categorise where and how method rationale supports communication during this process. The chapter also explores how the persons involved in method implementation view methods as consisting of modules in order to ascertain a suitable level of granularity for an improved method module concept.

5.1 Chapter Overview

This chapter involves the activity Empirical Grounding Posten IT: E-Gov as found in Figure 4. Input for this activity is primarily the method rationale
definition (called MR Definition in the model) as this concept functions as an analytical lens during my interpretation of what is going on during the implementation phase. This lens will give me a focus during my interpretations and give me direction in my analytical work. Putting the concept of method rationale into action like this also gives me a chance to test the viability of the concept in a research setting and a possibility to change any conceptions about the fundamental design.

The second input for this chapter is the method module notion (called MN Notion in the model). The MN Notion is an idea that systems development methods need to be modularised in a way that would make sense to method users and method researchers. The starting point for this idea is the analysis conducted in chapters 3 and 4 where I conduct an analysis of previous attempts to modularise systems development methods and formulate initial requirements for a method module concept capable of carrying the dimension of method rationale during application in all three method usage spheres (The method engineering sphere, the method-in-action sphere, and the method reflection sphere). In this sense, this chapter plays the role of setting an initial level of granularity for the improved method module concept. This was done by an analysis of how experienced method users think and talk about method parts and why. Thus, the primary output from this chapter is the level of granularity for the method module.

5.2 Interpretative Research Approach

This case study was conducted with an interpretative research approach (Walsham, 1995) since my focus during the case study was to explore and understand the communication potential of method rationale during method implementation. The exploratory approach undertaken focused mainly on communication through project documentation, and verbal accounts during meetings and seminars. Some supplementary interviews with the consultants also took place.

In this chapter I explore the communicative role of method rationale in the tasks suggested during the project assessment phase. This concept will be in the centre of attention of my analysis (Dyer Jr and Wilkins, 1991). This means that I will focus on and describe statements and actions that can be considered being about method rationale. In order to build a strong foundation for my analysis I
will have to present a rich picture of the case so that the reader can follow my interpretations and understand where the conclusions come from.

During the entire Posten IT case study, three researchers were involved. Our primary task for the researchers was to assist with methods and techniques for the task of adapting the RUP for Posten IT. In total we had meetings with representatives from Posten IT at nine occasions. The timeline of the case studies from Posten IT spans between 2002 and 2003 as depicted in Figure 5.

In chapter 2, there is a description of Eisenhardts’ (1989) distinction between three types of theory use in case studies. They can serve as a way to facilitate an initial guide to design and data collection, as a phase in an iterative process of data collection and analysis and, last but not least, as the resulting research product. The concept of method rationale was used as a theoretical descriptive framework that guided data collection and analysis (Patton, 1990). Subsequently, this case study will follow the second use of theory according to Eisenhardt (1989) as the case is a way to put previous theoretical findings into action, and a willingness to let them evolve or change in an iterative fashion.

This is done by application of the method rationale definition as an analytical lens as I analyse the communication regarding systems development methods among the consultants and the involved development team. The fundamental empirical base for my analysis can be found in documents produced by the consultants during their assessment tasks. These documents have been analysed conceptually and compared with the method rationale definition. Any statements regarding methods as carrying knowledge has been taken into consideration. This means that I have interpreted the empirical data in the light of method rationale. This definition, as stated in chapter 4, has functioned as a theoretical descriptive and analytical framework in accordance with Patton’s (1990) suggestions.

In the analysis in the end of the chapter, where I return to the notion of the method module, I also follow Patton’s approach. In that section the idea that the method needs to be modularised is compared to how experienced method consultants intuitively think of methods when they are divided into more manageable parts or modules. An idea with an improved systems development method concept for modularisation is that it should have a foundation in best practices and professional know-how of people who work with systems development methods on a daily basis. Only by ensuring that the concept has this foundation, it will be possible to design a method module concept that has a chance of actually being used. A module concept that seems intuitive and with a foundation in practical knowledge would also be beneficial for EME research in all method application spheres.

The assessment of the E-Gov project had already been initiated by the time we got involved in the research project. As a consequence, a lot of aspects had already been covered by the consultants responsible for the RUP implementation. It is also important to note that all project specific tasks
concerning the E-Gov project were conducted by either the consultants or the Posten IT employees. My role during the E-Gov project was restricted to observing and evaluating their efforts.

5.3 Case Setting
The case study describes events that took place during a research project connected to a major downsizing project called E25 at the Posten AB. E25 was an attempt to reduce administrative costs by 25%. In line with this task, the systems development department (Posten IT), consisting of approximately 500 people at Posten IT, implemented RUP as a standard systems development method.

In order to achieve an overall efficiency improvement it was deemed necessary to focus on finding common platforms for Posten AB’s web based IT strategy. This called for a standardisation of the systems development processes, the test environments, and business development processes.

The choice of RUP as the standard systems development method stems from a need for the rigorous support for all sorts of development activities that comes with RUP. The sheer size of the RUP with its ambition to give support for all types of activities during a development project along with an array of available tools give the RUP this rigorous appearance. A few initial hand picked projects were chosen for the implementation phase; the E-Gov project, the PSU project, the E-skicka project, and the ELLA project. The rationality behind each choice was that the projects’ systems development teams already had something that could be regarded as a systems development method although not formally externalised and defined. These types of informal systems development methods showed that the individual project teams already aimed to standardise their development processes and that they were concerned with its maintenance. A project team of this kind would have an easier task of adopting a systems development method such as RUP since they would have experience in method support, although informal, but even more important, they could appreciate the support a systems development method could deliver. Another reason for choosing a particular development team could be that the project they were working on demanded a particular form of documentation or a specific level of detail for some reason. The implementation process was led by consultants from Rational Software.

5.3.1 The E-Gov project
The E-Gov project aimed to develop, deliver and host customer specific solutions for the public sector. The ambition was to provide support for electronic and physical message transfer to parts of the public sector who are not capable of hosting the solutions by their own. This could be, for instance, smaller municipalities. Examples of the services offered could be planning agendas, identification protocols, Payment service or access control. The E-Gov
project did not aim to develop a solution which would suit everyone. Hence, the project was divided into sub projects handling the requirements from and delivering to their respective customers.

5.4 Fundamental Problems Related to Organisational Change

Any organisation planning to adopt a systems development method must prepare the organisation by creating an awareness of the systems development method being implemented. The reasons for doing so are obvious. People in general are not particularly willing to embrace change that they do not understand. Changes in their everyday life or in the way they work need to be understood. Only then can people in the organisation overcome their initial fear of failure (Carlin, 2006). Our minds tend to imagine worst-case scenarios, something psychologists call “awfulising”. Awfulising is classified as an irrational thought process (Davies, 2008) and as such it would be difficult for a developer to internalise a systems development methods’ underlying rationality when these irrational thoughts block rational reasoning. The reasons for awfulising to occur can be found in human evolution. We have a tendency to overestimate dangers and risks because they were life threatening through out much of human history. If we where not careful enough, we could be eaten by a lion for instance (Carlin, 2006). The existence of these thought processes is well known in psychological research and is a considerable difficulty when it comes to organisational change (Miller et al., 1994; Piderit, 2000; Fugate et al., 2008; Davies, 2008). Awfulising stands in direct conflict with the concept of method rationale since the concept assumes that method knowledge only is possible when the private rationality is in resonance with the public rationality provided by an external systems development method (Russo and Stolterman, 1997). To overcome awfulising it is therefore important to give the organisation a possibility to grasp the rationale behind the method implementation process. Planned goals must be communicated to the developers in the organisation so they will be able to understand the planned benefits and why the method implementation is necessary (Bergström and Råberg, 2004). Without explication of the method rationale the developers will not be able to appreciate the support the implemented method will provide because they will not be able to compare it with their own rationalities.

Even though the developers in the organisation might understand the rationale provided by the method and realise that the rationalities coincide, there might still be resistance to change. The reasons for this have been well understood by psychological research (Miller et al., 1994; Piderit, 2000; Fugate et al., 2008). Thus, it does not necessarily always have to do with the possibility to achieve rationality resonance. More common are reactions connected to the fact that people in general dislike alterations in their current work situation (Pulakos et al., 2000; van Dam et al., 2008). Another common reason is that some people
are not confident about their capabilities to perform in a changed work context (Parker, 1998).

The employees’ reactions to the change processes are therefore considered as critical for the success of change processes. As there seems to be a natural resistance to change in most organisations according to the research presented above, it is imperative that the method implementation process gains support and trust from the employees. An organisation must be willing to change for an implementation process to be successful. To overcome the initial and very natural reactions of resistance in the Posten IT case the persons responsible for the implementation process had to tread very carefully in their efforts to prepare the development organisation. They followed the process recommended in “Adopting The Rational Unified Process – Success With the RUP” (Bergström and Råberg, 2004). Not surprisingly, since Råberg was a consultant during the implementation process. Important tasks connected to organisational preparation during this process are described below.

5.5 The Role of Method Rationale during Method Implementation

During the analysis I have focused on important areas where method rationale plays a vital role during the initial stages of method implementation. The following section is devoted to a description of the role method rationale plays in these areas.

5.5.1 Creating Awareness of RUP

During this task the consultants started very early by letting the developers and other persons that would be affected, get acquainted with RUP very informally and without using any “big words” or technical terminology. By doing this it is easier to highlight general benefits from the planned implementation process. This is done by answering a lot of preconceived questions that might, or probably will arise. Examples of these types of questions could be “What is RUP?”, “What are its benefits?” or “How do we make it happen?”. By discussing these topics together the development team can get a general feeling for how RUP would contribute to their work situation. It is also important to address typical drawbacks and be fair and state that the process of adopting a systems development method as RUP is not an easy task. Developers who are used to develop systems through informal development methods are sometimes a little bit scared by large methods such as RUP since they believe that they will have to draw a lot of diagrams instead of writing code. This attitude is illustrated by the following quote from one developer at Posten IT. “UML is boring!” This quote comes from a group discussion with developers at an early stage of the implementation process. In order to change this developer’s attitude towards RUP the consultants had to focus on and highlight the benefits of RUP so that he would understand the role the diagrams play in RUP and how they could contribute to the overall success of the projects. Arguments that explained how
diagrams are related to other parts of the process are ways to communicate the benefits of a systems development method to a hesitant developer. It should be noted that this attitude towards RUP was not the common stance among the developers. In particular, the people involved in the E-Government project were enthusiastic and expressed an eagerness to start right away.

The key word during this phase is communication and it is important that the communication truly is two-way communication. The consultants at Posten IT took the developers anxiety seriously and made sure that they would have a say in how things would be done. This way trust was built and RUP did not appear as intimidating as it initially did to some parts of the organisation. Creating awareness of RUP is about communicating the overall rationality of RUP and letting the developers start to understand how this rationality can contribute to their projects. It also gives the developers an opportunity to judge whether the rationale behind RUP is something they could appreciate. In other words, letting the developers think about whether they can see that rationality resonance could occur.

An example of this could be a developer who experiences difficulties in consolidating a set of artefacts that could describe the sum of software requirements of the system. According to the developer’s rationality, this is something s/he would like to see support for. The persons responsible for the RUP implementation process can then choose to communicate the rationality behind the activity Detail the software requirements, an activity that supports this complicated task in RUP. This activity is basically an instruction for how to create an overview over the sum of requirement artefacts that have been created. The developer can then judge whether s/he can find the rationality behind this activity to be supportive in his/her daily work situation. In other words, s/he is making an estimation of his/her possibilities of reaching rationality resonance in the future by using this activity. At this point in the implementation process the exact details of how the activity creates this result is not necessarily covered since this phase just is about raising awareness about RUP as something that could be beneficial for the developers. Internalising the overall method rationale of various method components is sufficient for this estimation process.

5.5.2 The E-Gov Project Assessment

Before any method implementation can begin you will need to ascertain the level of maturity in order to find suitable pilot projects. These projects will be the key factors in the implementation of RUP in an organisation. Finding these pilot projects is difficult and requires analysis of the organisation in order to find the best suited projects. It is important to choose projects that have the best chance of success in order to motivate other development projects in the organisation. The assessment phase is an important step during a method implementation process. By assessing the current development organisation it is possible to get a clear picture of the current system development situation. In
some cases an assessment report may come to the conclusion that it would not be advisable to implement a new systems development method at all at the time. The reason could for instance be that the organisation is too immature. Every conceivable aspect (e.g. testing, project management, requirements, design, etc.) of systems development is analysed and compared to known best practices in the software industry (Bergström and Råberg, 2004).

The project assessment aims to determine the status of the development processes in order to ascertain which parts of RUP should be implemented, when, and in what order. During the E25 project, a number of projects were assessed, although only the E-Gov project and the PSU project will be covered in this dissertation. The PSU project will be covered in chapter 7. The E-Gov project was assessed using the guidelines in Bergström and Råberg (2004) and is a typical example of how these tasks are normally carried out. In the next section I will address the aspects of this standard procedure where communication of method rationale plays a vital role and how.

The assessment of the E-Gov project had already been initiated by the consultants at the time we got involved with the method implementation effort. This means that the assessment phase basically followed the consultants’ regular process for this task. Again this means that they followed the recommended steps as suggested in Bergström and Råberg (2004) using the templates suggested there.

The project assessment involves many perspectives. The document template used for this task contain headlines that let us know what aspects are considered important to focus on in order to ensure a well informed decision in choosing projects for RUP implementation. The document has a layout that resembles RUP documentation and covers general aspects such as budgets, personnel, stakeholders, project team structure, system description and other general information about the project. This information is considered important since not only team knowledge is important when choosing a project for RUP implementation. An assessment that only focuses on the individual and total experience and the rationalities of the affected team members might yield a picture that shows that the project might be well suited for the endeavour. However, it might be the case that other factors would prove the project to be unsuitable. It might be the case that the project is coming to an end in the near future or that it might be subjected to future budget cuts. It might also be the case that certain stakeholders have requirements that might conflict with the implementation process. If the project has the role of a sub-contractor there might be regulations to what the project should deliver to their customers, especially when it comes to things like system documentation. It might also be the case that management has decided that the organisation should use a specific project management model. This was, for instance, the case for the projects in Postal Service’s IT department. They were given instructions that all projects should apply an internally developed project management model called
ATUP. This means that the RUP discipline Project management should not be implemented and ATUP would be used instead.

This is an example of how rationalities might conflict during an assessment phase. The method rationale offered (commonly best practises on how to run a project) in the Project management discipline is something that could help an organisation to improve their abilities to drive their projects. If there is a policy that dictates that another project management model should be used, the persons responsible for the method implementation process must adhere to the fact that their own rationale (their own interpretation of how to best run a project) is not possible to match to the one offered by for instance RUP. In other words, policy decisions have made a situation where this specific rationality resonance would be unattainable. Instead they had to judge their possibilities to reach rationality resonance by using the method rationale provided by ATUP. In the case of the E-Gov project the persons involved did not find a problem in this case since they did not have a specific project model at all. Using any of the suggested models would have improved the situation considerably as the only project management activities that had been implemented was the use of a web platform for consolidating documentation, plans and protocols. Since ATUP was to be used together with RUP the consultants realised that they would have to analyse the interface between the project management discipline and the rest of the RUP to understand how it contributes to the overall method rationale offered by the RUP. Changing RUP’s project management discipline for ATUP also meant that the method rationale provided by ATUP must be analysed and understood in order to find if and how any changes must be made in the interface. This is important; otherwise certain activities undertaken in ATUP might not be useful in other parts of the project. More importantly; as the project management discipline in the RUP takes all activities and documentation produced during a RUP project into consideration, there is always the risk that ATUP misses to handle certain aspects of RUP. The iterative development perspective emphasised in RUP might for instance be difficult to handle if ATUP does not provide method rationale for this project planning activity. Again, we see an example of how the public method rationale provided by a method (RUP, ATUP) is analysed and compared to the private rationality of the persons involved in the process.

A major part of the project assessment document is devoted to interviews with the project members. The interviews focused on a number of aspects in order to investigate the team’s possibilities to be suited for RUP implementation.

5.5.2.1 General Method Knowledge

The consultants used interviews to ascertain the current development method maturity status. The questions focused on how the project used development methods today and the individual skills and previous RUP experience the developers possessed. The E-Gov project primarily used a waterfall model
which gave some direction when it came to what to do and when. However, they
did not have any sufficient support for the activities during the waterfall model’s
different phases. This indicated that the project team had an ambition to control
the development process but did not know how to do this satisfactorily and with
enough detail. Another way to put it is to say that the developers had not found
or adopted an outside rationale for these specific activities. Their own private
rationale guided them during their everyday labour but they did not try to match
it with any public rationale provided by a systems development method in an
extent that would be sufficient. Hence, states of rationality resonance were
difficult to achieve.

5.5.2.1 RUP Knowledge
An important task during the method reconstruction phase was to find out if any
of the developers had any previous experience with the RUP. If so, then the
method implementation process would be less cumbersome since at least some
of the developers would have had some sort of prior contact with the method.
Needless to say, the prior knowledge would help them to grasp the RUP more
easily. Most of the developers in the project had very little experience with the
RUP. Some had studied the RUP in beginners’ courses but had not used the
method in actual projects. One of the developers was regularly involved in
different projects as a RUP mentor in the requirements and test disciplines.
This means that parts of the team did have some knowledge about RUP but it
does not necessarily say that they had any understanding of the method rationale
provided by RUP. They may have some insight but since they did not have any
experience in using RUP it was difficult to ascertain if they fully understood
what RUP could provide in terms of method rationale.

5.5.2.2 Attitude towards RUP
In any method implementation endeavour it is important to pick implementation
projects that would have the best chances to succeed. In the Postal Service case
study a small number of projects where chosen with the intent to be success
examples for the rest of the development organisation. The reason for this is to
avoid unnecessary negative attitude towards the implementation project. As
described earlier in the chapter, any organisation change is important to anchor
among the affected people since organisations naturally have a fear of changes,
especially if the directives come from top management.
In other words the people involved would have an easier task if they can
judge the possible benefits of an implementation process if another project in the
same organisation has already made it before. The attitude towards the RUP
implementation project is closely related to method rationale since any signals
sent from a pilot project indicate success or failure. If the project is running well
and the developers involved state that they have a much easier task to deliver in
time than before they are not only implicitly telling listeners that they appreciate
the method rationale offered by RUP. Telling others that they have improved
their development process does also send signals that the team was able to achieve rationality resonance. Even though they are not familiar with the concept of method rationale they are able to approximate a co-workers possibility to work more efficiently by the help of the newly implemented method and that line of thinking follows the same mental structures as when talking about method rationale.

In the E-Gov project, most of the developers where eager to be selected for the pilot project since they felt that they would benefit from RUP. The only exception was that one of the developers stated that “UML is boring!” although s/he could appreciate other parts from RUP as beneficial. In the UML case it was clear that the developer in question did not normally use any visual modelling techniques for the same reason. In this case rationality resonance had failed to occur for some reason. This attitude would prove to change the RUP implementation plan dramatically as we shall see later on.

5.5.2.3 Current Tasks and Method Support
During the interviews the consultants asked the project team about their current situation in terms of method support. The ambition was to identify problems in the current informal development process in order to ascertain which parts of the RUP would be most critical to implement first. The questions focused on a central perspective in the RUP called best practises. The best practises are a way to pinpoint the main benefits of the RUP in easy to understand slogans: 1. Develop software iteratively, 2. Manage requirements, 3. Use component-based architectures, 4. Visually model software, 5. Verify software quality, and 6. Control changes to software. In the consultants’ interview with the project manager, she stated that they did not have any formal process or model for governing the project as a whole but they did have regular meetings and a list of things that needed attention. There were no plans for any iterations. She identified a need for management of the project’s documentation and some sort of function to keep track of changes in requirements. The consultants’ conclusion regarding how the E-Gov project could benefit from the RUP followed this type of reasoning:

1. What are the problems in the current development situation?
2. What type of support could handle the problems?
3. What in the RUP can provide the desired support?

The second question means that the consultants had a focus on method rationale. The support they are seeking to find are goals (method rationale) provided by certain parts of the RUP. An example from the interview with the project manager would be that the problem of not being able to keep track of changes in requirements is something that needs to be solved. Thus, the method rationale for addressing these issues are explored in the RUP and discovered in
the discipline named Configuration and change management. Also the need for a project model is taken into consideration. Problems with planning for iterations must be dealt with. Hence, some kind of public method rationale possible of supporting this problem is required. The consultants are aware that the Project management discipline in the RUP could handle the issue satisfactorily. However, they are aware of the fact that ATUP is supposed to be adopted as project management support. For that reason, the project management discipline is omitted and ATUP is chosen instead even though the consultants are aware that they probably will have to make some alterations in either (or both) ATUP and the rest of the RUP that is being implemented.

Another aspect that was discovered during the interviews was the lack of support for gathering requirements in the form of proper documentation such as use case models or use case specifications. Sometimes they were handed use case specifications from their customers but they had not any formalised way to handle these nor did the have any regularity in producing documents by their own. Some members of the team sometimes modelled or wrote use cases by themselves. They did so at times when they felt that they had to get a clearer picture of what they where supposed to do and used Microsoft Visio or Microsoft Word for the task. These types of reasoning show that the developers could appreciate the public method rationale provided by the use case method component and use it in a situation where rationality resonance could occur. A problem with this situation was that the documents were almost only produced to solve problems on an individual level. Seldom were documents passed on to other parts of the project, nor were they usually managed, updated, or regarded as very central. In essence, they were just individual notes before implementation. This showed the consultants that the team needed some sort of public rationale for handling requirements and hence found the solution in the RUP discipline known as Requirements. By implementing the requirements discipline the E-Gov project would benefit from the formalisation of producing and managing the documents needed for the projects.

The requirements discipline is closely related to the best practice Visually model software where the RUP suggest that UML should be used for not only requirements but also for analysis and design. Of course there is a corresponding discipline in the RUP called Analysis and Design, however since these types of UML documents were frowned upon by the development team (particularly one developer as described earlier), the consultant thought it best not to impose this particular discipline at this time. Another way to put it would be to say that they estimated that the possibilities for the team to successfully reach rationality resonance during analysis and design with the help of method components from UML were very unlikely.

The same types of problems were found when the interviews focused on the use of component-based architecture. The developers were reluctant to comment code and document the system. The rationale for these types of tasks is available
in the RUP but once again the attitude to UML and documentation among the programmers proved that a RUP implementation to handle these specific problems would be difficult to impose since the possibilities were slim for creating a development situation where true rationality resonance would occur.

A problem area, indicated in the interviews, was the need to handle software quality. The team had a member who had knowledge of the Test discipline from the RUP and showed the consultants that they not only had pinpointed a problem and could plan a solution, they could also be somewhat certain that an implementation of the test discipline would be successful.

All in all the consultants decided that an implementation of the RUP disciplines; Requirements, Configuration and Change management, and Test were appropriate. The discipline Project management would be omitted since ATUP would be used instead. This means that the disciplines Analysis and Design, Implementation, Deployment and Environment were not chosen. Deployment was never really considered since most parts of the system had already been delivered and the E-Gov project was more of a maintenance project at the time of the case study. The environment discipline is most commonly implemented at an organisational level since it involves activities related to organisational preparation, i.e. what the consultants were there to do. The analysis and design discipline and the implementation discipline were not chosen for reasons described above.

The consultants also chose to only implement parts of the selected disciplines in the beginning as the E-Gov projects team members had a low level of method maturity. Introducing a new systems development method to a project means that the persons involved must be given a chance to internalise the method rationale offered by the new method in a balanced pace, otherwise there is an apparent risk that it all would be perceived as too confusing.

5.6 Method Rationale - Lessons Learned

The analysis of the project assessment phase in the E-Gov project involved viewing the actions taken by the consultants through the lens provided by method rationale. More concretely, this means that I have viewed their conceptions about what a method is and their views on how methods support method users in the light of the method rationale definition from chapter 4. This has been done with the help of the figure describing rationality resonance from chapter 4, which can be seen below.

During the course of project assessment the consultants viewed the RUP as a public rationality with the capability of providing the individual developers solutions for how they should work. At the same time they addressed questions to the involved developers in order to understand how they usually worked, what they felt about methods in general and about the RUP in particular. This indicates that they addressed and focused on the developers’ individual intentions and needs, i.e. the developers’ private rationality. In essence, what the
consultants did was to understand how the public rationality provided by the RUP could harmonise with the individual private rationality of the developers. Hence, they were assessing strengths and problems connected to the possibility to achieve rationality resonance.

Going back to the ontological framework for method rationale presented in chapter 4, asking questions of this kind focuses an understanding of how the developers perform in the method-in-action sphere as well as their ideas and needs as formulated in the method reflection sphere. As the RUP is considered to be a public rationality, constructed and formulated beforehand and possible to convey, we also can cover the method engineering sphere. This not only shows that the concept of method rationale is useful for addressing method research on all three levels, it also shows that the definition, presented in chapter 4, is possible to use for this type of research.

So far, this does not give us any reason to alter the previous definition of method rationale.

### 5.7 Weaknesses in Current Approach for Project Assessment

During the project assessment phase, the consultants tried to create an overview of how the developers worked today. This involved an attempt to reconstruct the existing informal development process into an externalised systems development method. By externalised, I am not referring to a systems development method that can be conveyed or taught to others. The externalised method played the role an object for comparison in relation to the RUP. In order for the consultants to have a chance to compare the informal systems development method that was in use with the method being implemented, they

![Figure 20 Rationality Resonance (From chapter 4)](image-url)
had to try to understand as much as possible about how the work was conducted. Hence, all questions addressing all sorts of aspects of systems development business.

The questions most clearly focusing on a reconstruction process were the ones that addressed current tasks and method support. The interviews showed that the development team had a very varying level of method maturity which had consequences for how much methods were being used during daily work. Some of the developers had taken beginners’ courses in the RUP and one team member was regularly involved in different projects as a RUP mentor in the requirements and test disciplines. The interviews also showed that some of the developers used method when they modelled software visually, even though they only did so to solve individual problems instead of team problems.

In the end the consultants had formulated an opinion about which disciplines from the RUP to implement. They chose the disciplines Requirements, Configuration and Change management, and Test to be the correct disciplines to be implemented first. All of this made perfect sense to the consultants. One of the consultants later explained that these disciplines typically were chosen as they often represent what seems to be missing from a RUP perspective. A problem with this view is that it is likely that consultants working with these premises might have had a strong preconception about what needed to be done. It might just be the case that they only did what they usually do. Let us go back to how they actually did things. The consultants took a very clear problem oriented view as a starting point. This means that a real reconstruction of how work was being conducted never really took place. The consultants were more focused on finding discrepancies between how the developers worked with what is being out forward in the RUP. They never really tried to understand any of the potential strengths in the informal systems development method in the E-Gov project. They were considerably keener on finding problem areas for solutions they might have already had decided upon. The chosen approach for finding out which part of the RUP to implement first certainly indicates this.

5.7.1 Choosing a Level of Granularity for the Method Module Notion

During the time the decisions were being made about what disciplines of the RUP that would be implemented first, we interviewed the consultants about how they where going to implement the disciplines in more detail. Strangely enough, it became evident that they were not going to implement any disciplines at all! Rather, they were going to implement various sub-sets of the chosen disciplines. Of course, this made us curious as to which parts that would be selected and how they viewed the different parts to be separated. The presentation given to us by the consultants revealed that the chosen size, or level of granularity, of the method part could be used as a requirement for the new method module concept. In particular since they explained that it was a natural, or at least proven, way of dividing the RUP into more manageable parts.
The level of granularity used by the consultants in the Posten IT:E-Gov project assessment phase corresponded to the Diagram level of granularity according to Brinkkemper et al’s (1999) method fragment concept.

There are five possible levels: method, stage, model, diagram, and concept. The method level can not be used as a module notion since it does not involve any modules. It covers the entire systems development method, i.e. the RUP or OMT (Rumbaugh et al., 1991). The stage level is too large to actually be able to carry any knowledge of how the method is supposed to, or could, be used. An example of this is the consultants’ immediate division of the chosen RUP disciplines, which correspond to the stage level, into more manageable parts. The model level addresses a total given perspective on a development issue, for example the Use case model in the RUP. The use case model is really just a collection of various diagrams and documents that capture all system interaction and is also typically too large to be considered as appropriate for a module notion. The concept level addresses the conceptual building blocks that are used while using a systems development method. An actor is an example of a concept describing a role someone has when interacting with the system and can be considered too small for a module notion. Since a systems development method is not something that exists for its own sake but rather a way of consolidating development efforts and to communicate results to other people, the concept level can not be used as the level of granularity for the method module notion. Basically, a method user has not consolidated his/her efforts by only finding an actor during analysis. S/he probably has not even produced an intermediate result interesting enough to communicate to other developers either.

A result worth communicating, in the sense that it could be the starting point for someone else involved in the development process, comes first at the diagram level of granularity. Only when something has been produced on this level it has a meaning that is independent and possible to grasp when communicated. It is important to know that the diagram level of granularity does not necessarily mean that everything that can be considered to reside on this level must be in the form of a diagram such as expressed with UML notation. It could very well be an artefact in the form of a textual document. The deciding factor is that the artefact has a foundation in the conceptual building blocks that constitute the method and that the artefact helps to shed light over the perspective expressed in the corresponding model level.

The discussion above can only lead to the conclusion that the diagram level of granularity as suggested by Brinkkemper et al (1999) will be the most appropriate level of granularity for the new method concept.

5.8 Concluding Discussion

Going back to the model describing EME in Figure 14, the case study mostly resided in the method-in-action sphere as it focused method usage. However, large parts of the research were conducted with a focus on method reflection.
Hence, research also took place within the method reflection sphere, typically when aspects that scrutinised method usage and its effects came into focus. The case study covered the initial stages of a method implementation project and shows how method rationale plays a vital role when addressing the tasks connected to preparing the development organisation to change and to assess the current status of the project. All assessments task are centred around communication of practices, conceptions and rationalities as the persons responsible for the task tries to understand the possibilities and limitations of the organisation before any implementation takes place.

An organisation is typically reluctant to change and attitudes towards any organisational change might stir up conflicts. Knowledge and understanding of this organisational inertia is created through communication about the project where method rationale is used to point out what can be gained. Communication of the underlying and overall method rationale of the method being implemented is something that involves development teams as well as management. This gives a goal oriented focus and facilitates the process of understanding why certain changes in the development organisation is required or advised and why it cost money. Management needs to understand the monetary benefits of the implementation. Especially so, since it often is the case that the implementation project was initiated in order to improve production and lower costs. The development teams need to understand how the implementation project can help them to perform better, improve their productivity and manage their development process. Understanding the goals of the method being implemented enables the affected persons to compare the public method rationale of the method being implemented, with their own private rationalities. It gives them the opportunity to think about their possibilities to reach rationality resonance once the method implementation project is finished. The persons responsible for the implementation project have the roles of communicators of method rationale while they are collecting, comparing and evaluating the rationalities of the management, the developers and the chosen systems development method in order to find a suitable solution for how the project should run. This involves finding pilot projects, success stories and carefully selecting method components to be implemented in the right place at the right time. They must find a balance between the total sum of public method rationale the base method offers and the needs expressed through private rationality coming from the development teams. Otherwise, the development teams might end up in a state where they “awfulise” the organisational change the implementation project aims to achieve. Implementing a systems development method in an organisation means that method rationale is communicated back and forth between the people involved and it is what gives it meaning and sense.

Based on the discussion presented above, I find no reasons for changing the method rationale definition at this point.
An important lesson during the Posten IT: E-Gov case study was that the method reconstruction phase possibly could lead to biased decisions. Some aspects of what the consultants did indicate that they might have already decided what to implement and in what order. In order to truly apply a method rationale perspective through the entire project assessment phase and really give the developers a chance to have a say in how their own rationality resonance might possibly be achieved, we must find a better way for understanding informal systems development methods.

Another aspect concerning the choices the consultants made about what to actually implement from the RUP disciplines was that the level of granularity for the method module notion might be the diagram level according to Brinkkemper et al. (1999). This level of granularity not only corresponded to the consultants’ intuitive way of dividing the RUP into manageable parts it also made sense when its potential was compared to the other suggested levels; method, stage, model, and concept.
6. The Method Component Concept

In section 4.4 I presented an argumentation that there is a need for a method module notion capable of carrying method rationale. This is important to consider since method rationale actually must be considered to be the fundamental argument to why anyone should follow the line of action proposed by the method as well as the individual method modules. Separation and aggregation of a systems development method is something that is common within the field of method engineering (Brinkkemper, 1996; Odell, 1996; Harmsen, 1997; Ralyté et al., 2003; Karlsson, 2005). The concept of method rationale must be applicable on both complete systems development methods as well as on individual method modules. Thus, the concept of method rationale must be considered as being inherent in the method modules in order for them to be applicable in a separation or aggregation scenario. Otherwise, it would be difficult to understand the individual benefits that may come from using a specific method or method module in a given systems development project or why a certain method or method module might be considered unnecessary in another development project. I have previously elicited three initial requirements for such a module. This chapter is devoted to a formulation and design of a method module concept capable of incorporating these requirements.

- Possibilities to create equal theoretical and practical focus on method rationale in all method application spheres.
- Possibilities to express and describe all instantiations of method use.
- Possibilities to modularise a systems development method into consistent and coherent units, capable of giving feedback between the spheres and also serve as a common ground for EME.

These requirements will be analysed and reformulated into design goals in order to create a module concept that has potential as a theoretical foundation for EME. This module concept will be possible to apply in research in any of the method application spheres defined in Figure 14. Another aspect that has to be taken into consideration is that the module concept must be able to be used in what is considered to be traditional method engineering as well as in what I refer to as ISD research. Even though I take a starting point in method engineering in this dissertation, there must be possibilities to create knowledge transfer between the two fields of research regardless of which method application sphere the method research takes place in.

6.1 Chapter Overview

This chapter covers three conducted activities. The first conducted activity was an elicitation of design goals for the suggested method module notion. Input for this activity was the chosen level of granularity (Diagram level according to
Brinkkemper (1999)) from chapter 5 and the method rationale definition. The definition was needed for this activity since the method module concept is intended to be able to carry method rationale. Otherwise, it would not be useful for EME research in all method application spheres (see Figure 14 in chapter 4). The defined output from this activity was the design goals.

The following activity was the “choice of method components as starting module concept” In this activity, the author and a colleague conducted a conceptual analysis of the method fragments, the method chunks, and the method components. Inputs were the module notion and the level of granularity. The output was the method component concept as defined by Röstlinger and Goldkuhl (1994).

The third activity covered in this chapter was a “conceptual re-modelling of the method component concept”. In this chapter the original method component was subjected to another conceptual analysis and re-modelled. Inputs for this chapter were the method component, the design goals, and the method rationale definition.

A model describing the chapter details is depicted below.

Figure 21. Chapter 6 details

6.2 Elicitation and Formulation of Design Goals for an Improved Method Concept

To ensure that the requirements are taken into consideration when the improved method module concept is designed they need to be broken down into clear cut design goals that can function as a guideline for the following design process. In this section I will address the initial requirements of such a method module, elicited in section 4.4, and reformulate them into design goals. In order to reach a state where the design goals will have a chance to yield a method module concept capable of carrying method rationale I will have to take the method rationale definition into consideration. The same goes for the chosen level of granularity presented in section 5.7.1. This decision concerns how much of a method the method module concept should cover. I have presented empirical data that implies that a level of granularity corresponding to Brinkkemper et.
al.’s (1999) diagram level is appropriate since it covers enough of the method to actually carry knowledge of what the method user should do and why but at the same time be possible to treat as singular separated units. The diagram level of granularity also seems to be the intuitive choice of the method consultants in chapter 5.

The requirement that concerns ‘Possibilities to create equal focus on method rationale in all method application spheres’, raises a need for a design capable of expressing the dimension of rationality. Hence, Rationality must be a design goal. The method module concept must be able to carry and express its rationality dimension in all method application spheres in order to function as a common ground for EME and research that transcendent the method application spheres and enables knowledge transfer between research fields. The contention is to extend the field of method engineering in a way that would move the boundaries of what can be considered method engineering and at the same time give researchers typically belonging to the field of ISD an insight in how they can use theories and concepts traditionally from the engineering field in interpretative research. This first requirement also raises a need for a design goal considering equal treatment of the method module concept in all spheres and in all research. To ensure this, the module must be comprehensible and this yields a demand for a design that is internally consistent and coherent. This way the module can be understood as a stable entity with a predictable structure, with a level of granularity that is deemed necessary for enabling knowledge transfer through method rationale. This also means that the method module should be possible to grasp as a Self-contained entity.

The requirement that concerns ‘Possibilities to express and describe all instantiations of method use’ also raises needs for design goals concerning rationality, internal consistence and coherence, and self-contain ability. It also raises a need for a design goal that concerns the method modules’ Applicability. It must be possible to map the method module concept onto any systems development method in any relevant situation. Furthermore the requirement also raises a need for a design goal that addresses the possibilities to modularise (top–down) or to engineer, or otherwise build, (bottom–up) a systems development method. This means that the method modules should be able to connect to each other in a way that ensures that their inherent method rationale is visible and how the method rationale of each module can contribute to the overall goals of the systems development method or a specific systems development project. The requirement that concerns ‘Possibilities to modularise a systems development method into consistent and coherent units, capable for giving feedback and to enable knowledge transfer between the spheres and also between the two fields of ME and ISD’ is applicable in all design goals already stated. It also captures my ambition to design a method module concept that can contribute to EME and enable knowledge transfer and feedback between spheres and method research fields by taking the elicited design goals into consideration.
It is my contention that these elicited design goals will yield a construct that will be proven useful, meaningful and capable in this sense. The design goals can be summarised as follows:

- **Self-contained**: It must be possible to treat the method module as a self-contained part of a systems development method with regard to the guidelines that describe the deliverable and the process of producing such a deliverable. This way a method module can be modelled out from a systems development method or act as a building block when constructing new methods or during method integration.

- **Internal consistence and coherence**: The method module must be perceivable as an internally consistent and coherent entity. Another way to put it is to say that the construct should be stable over time and without lose ends. As a result the method module will be perceivable as meaningful. The principle is an ambition to create congruence amongst method modules and homogeneity in systems development methods.

- **Rationality**: Actions are performed and prescribed for reasons. Thus a method module must have an identified target state, a purpose for its existence. The arguments are motivated by the modules inherent method rationale and address the goals and values that the method module realises.

- **Connectivity**: The method modules must be possible to connect with each other. Each method module, viewed as part of the systems development method, should contribute to a chain of goal achievements adding to the overall goal of the specific project. Thus method modules must have the capability to express how they are connected to each other and how these connections can contribute to the overall goals linked to a specific project.

- **Applicability**: It should be possible to map the method module construct onto any existing systems development method in any situation.

### 6.3 Choice of Starting Method Module Concept

You often hear the phrase that “You should not re-invent the wheel” with the meaning that it is not advisable or necessary to begin realising everything from scratch. You will often find that someone else has already thought in similar ways about the things you have on your mind. This is naturally also the case with module concepts in systems development methods. As I have shown in chapter 3 there are different ways of dividing a systems development method into separated parts. In method engineering the most prominent examples can be
found in the method fragments (Brinkkemper et al., 1999; Harmsen, 1997) or
the method chunks (Rolland and Prakash, 1996; Ralyté et al., 2003). Another
example can be found in Röstlinger and Goldkuhls’ method components (1994).

In this section we will bring in the notion of the method module into already
existing conceptualisations of systems development method modules and
compare them with my two other knowledge objects; the chosen level of
granularity and the design goals (see Figure 21).

6.3.1 Analysis of the Method Fragment Concept

The method fragment concept as suggested by Brinkkemper et al. (1999) is
presented in chapter 3. In essence the method fragments aim to function as a
module concept for method engineering. As such, they are possible to define and
manipulate through the use of a method engineering language (MEL)
(Brinkkemper et al., 1999; Harmsen, 1997). Their idea is to construct a new
situation specific systems development method for each project with the help of
method fragments stored in a data base called method base.

   *Self-contained: It must be possible to treat the method module as a
self-contained part of a systems development method with regard
to the guidelines that describe the deliverable and the process of
producing such a deliverable. This way a method module can be
modelled out from a systems development method or act as a
building block when constructing new methods or during method
integration.*

The method fragments are possible to treat as self-contained modules in one
sense since they have the possibility to classify fragments as being of either
product or process perspective. A method fragment that is large enough to
encompass both these perspectives can be regarded as self-contained. However,
there are no rules that say that every method fragment should have
corresponding perspectives in this sense, nor that every method fragment should
contain enough information for actually producing what could be regarded as a
deliverable. Remember, a method fragment can exist on many levels of
granularity and it is possible to regard a single concept or a symbol as a method
fragment of its own. In that sense the method fragments can not be said to
always have the capability to function as self-contained entities.

   *Internal consistence and coherence: The method module must be
perceivable as an internally consistent and coherent entity.
Another way to put it is to say that the construct should be stable
over time and without lose ends. As a result the method module
will be perceivable as meaningful. The principle is an ambition to*
create congruence amongst method modules and homogeneity in systems development methods.

The main idea with method fragments is to create new systems development methods for individual projects. This is not always possible since method training often takes time and the effort of developing unique systems development method for each project might be too costly. In order to function as building blocks for this task the method fragments have the property of being extremely flexible. Whereas many method adaptation efforts (Fitzgerald et al., 2003; Karlsson, 2005) tends to omit or include activities in order to find a well functioning development process, the proponents of method fragments are willing to invent completely new method activities and deliverables (Brinkkemper et al., 1999). This leads to unstable method modules that do not meet the requirements set in the design goals as they can not be considered to be stable over time. They can not be regarded as being without loose ends as each fragment can be of different size ranging from a complete method to a singular concept. Thus it is possible to have a method fragments that does not make any sense when it is taken out of context. Furthermore, since the fragments can be of varying size they do not meet the requirement of being congruent thus making it difficult to create a homogenous systems development method out of building blocks that look very different from each other. In my analysis I have chosen to define a level of granularity for my method module concept corresponding to the diagram level according to the method fragments. Another way to interpret this would be to say that my argumentation so far only can acknowledge method fragments that reside on this level as being fit for consideration.

Rationality: Actions are performed and prescribed for reasons. Thus a method module must have an identified target state, a purpose for its existence. The arguments are motivated by the modules inherent method rationale and address the goals and values that the method module realises.

The knowledge dimension of systems development methods are completely forgotten or ignored. Method engineers usually assume that a method module is sufficient as a guideline for various development endeavours. Actually understanding how the situational method is to be used or what goals it fulfils are of lesser importance in the field of method engineering. Subsequently, the method fragments do not cover any of the underlying goals or values that motivates why anyone should actually do what is suggested in the method fragment. There is no rationality dimension, what so ever.

Connectivity: The method modules must be possible to connect with each other. Each method module, viewed as part of the
systems development method, should contribute to a chain of goal achievements adding to the overall goal of the specific project. Thus method modules must have the capability to express how they are connected to each other and how these connections can contribute to the overall goals linked to a specific project.

This design goal can be considered as partially met as it is possible and proven that a complete systems development method can be constructed from method fragments. However, the way of how the method fragments are connected to each other differs from how it is suggested in the design goal. The design goal states that the method modules’ ability to connect to each other should be based on a notion of an existing chain of goal achievements. As the method fragments do not acknowledge the underlying method rationale, connecting them with the help of goals is not possible.

Applicability: It should be possible to map the method module construct onto any existing systems development method in any situation.

This design goal can be considered to be met. As far as I can conclude there would be no problem of describing any systems development method with the help of method fragments.

6.3.2 Analysis of the Method Chunk Concept
The method chunk concept is another method module concept that comes from the field of method engineering. A chunk is an attempt to create a module concept with a tighter coupling between process and product. In the chunk concept these two types of method fragments are integrated in one single module (Rolland et al., 1999; Ralytė, 2002; Ralytė et al., 2003). In this way any systems development method can be regarded as a set of method chunks on different levels of granularity, thus a whole method can be regarded as a chunk in the same fashion a method fragment can constitute a whole systems development method.

Self-contained: It must be possible to treat the method module as a self-contained part of a systems development method with regard to the guidelines that describe the deliverable and the process of producing such a deliverable. This way a method module can be modelled out from a systems development method or act as a building block when constructing new methods or during method integration.
As method chunks can be defined on different levels of granularity, they have the same fault in comparison to the design goals as the method fragments. The method chunk always has a corresponding process fragment for each product fragment and in that sense, they can be said to be self-contained as they provide both a product and the process of producing that product. However, since a method chunk can be of any level of granularity it could yield method modules that does not appear to be self-contained as they might be difficult to interpret out of context.

*Internal consistence and coherence:* The method module must be perceivable as an internally consistent and coherent entity. Another way to put it is to say that the construct should be stable over time and without lose ends. As a result the method module will be perceivable as meaningful. The principle is an ambition to create congruence amongst method modules and homogeneity in systems development methods.

The main point of method chunks is to create new systems development methods for each individual project, just like the method fragments and with the same difficulties as that approach in terms of cost. The undefined level of granularity also lead to a situation where a method chunk can not always be regarded as consistent and coherent as they could be too small for carrying any deeper knowledge about systems development. Much like the method fragments, a method chunk can be modelled out until it does not have any meaning as it is out of context.

*Rationality:* Actions are performed and prescribed for reasons. Thus a method module must have an identified target state, a purpose for its existence. The arguments are motivated by the modules inherent method rationale and address the goals and values that the method module realises.

Unlike method fragments, method chunks incorporate an idea of a target state or a purpose for their existence. The chunks have guidelines with corresponding intentions, which are the targets of the product parts that are produced during a systems development project. However, the intentions do not correspond to our notion of method rationale. They merely refer to the construction of something useful in a development project. The intention is described as a statement of something that should be done, i.e. *Find an Actor* but does not say anything about why this is important, what possible goals would be fulfilled, or why this is deemed as realising any values. In essence, the proponents of the method chunk concept consider a rationality dimension of methods but they treat it as something that is given, clear and uncomplicated.
Connectivity: The method modules must be possible to connect with each other. Each method module, viewed as part of the systems development method, should contribute to a chain of goal achievements adding to the overall goal of the specific project. Thus method modules must have the capability to express how they are connected to each other and how these connections can contribute to the overall goals linked to a specific project.

Chunks are defined from a process perspective. This means that, in comparison with the method fragment concept, the core of each chunk is a guideline to a specific process and its related product fragments (Ralyté and Rolland, 2001). The chunks are then used to construct new methods in a similar way method fragments are. In that sense the method chunks can be said to have met the design goal of connectivity. However, they lack the goal oriented focus set by the design goal. In comparison with the method fragments, the method chunks do have more in common with the requirement set by the design since they incorporate guidelines with corresponding intentions. But, since the actual goals of the intentions are hidden they can not be regarded as fully adhering to the design goal of connectivity.

Applicability: It should be possible to map the method module construct onto any existing systems development method in any situation.

Again, it seems that it would not be problematic to apply a method chunk perspective on any systems development method. In terms of applicability, the method chunk concept does not seem to have any problems.

6.3.3 Analysis of the Method Component Concept

In the field of ISD there are also conceptualisations of method modules. They are not necessarily defined for method engineering as they can also be used for understanding what a systems development method is and what it contains. Röstlinger and Goldkuhl (1994) argue for the use of method components (also presented in chapter 3) as a module concept for systems development methods. In comparison with the method fragment concept and the method chunk concept the method component has taken a standpoint when it comes to the question of granularity of method modules. The method component takes a starting point in a level of granularity corresponding to the diagram level in the method fragment concept. As such, they are in line what we are looking for.

Self-contained: It must be possible to treat the method module as a self-contained part of a systems development method with regard
to the guidelines that describe the deliverable and the process of producing such a deliverable. This way a method module can be modelled out from a systems development method or act as a building block when constructing new methods or during method integration.

The method components intend to combine stability with flexibility. The basic idea with method components is to use a rather high level of abstraction and structure the component’s content based on internal dependencies. In other words the method component is intended to be self-contained. Furthermore, a component is always constituted of concepts (what to talk about), procedure (what questions to ask), and notations (how to express the answers)(Goldkuhl, 1991). This gives the method components the ability to be regarded as self-contained since they contain enough in themselves to be perceived as meaningful.

**Internal consistence and coherence:** The method module must be perceivable as an internally consistent and coherent entity. Another way to put it is to say that the construct should be stable over time and without lose ends. As a result the method module will be perceivable as meaningful. The principle is an ambition to create congruence amongst method modules and homogeneity in systems development methods.

The method components have taken a starting point in a defined level of granularity. As a result from this they present a congruent picture of the method module concept. They are internally consistent as and coherent as they all contain aspects that clarify what to work with, how to work, and how to record the results. Since all method components have this structure they can present a coherent picture of how certain development activities should be enacted. Consequently, a method component can be modelled out of a larger systems development method and still be perceived as meaningful. The requirements for classifying a method module to be a method component, is whether or not it represents a coherent picture for handling a specific and delimited problem in a systems development project.

**Rationality:** Actions are performed and prescribed for reasons. Thus a method module must have an identified target state, a purpose for its existence. The arguments are motivated by the modules inherent method rationale and address the goals and values that the method module realises.
The concepts in the method components are the result of the underlying perspective of the method creator and direct the attention towards certain phenomena in a development context. The perspective also influences cooperation forms in which the method components are enacted and the frameworks in which method components can be integrated (Goldkuhl, 1994). This tells us that there is a point with each method component. It guides the method user towards certain goals for a reason. The perspective of a method component consist of categories, definitions, values and goals (Goldkuhl, 1994). Hence, the method component contains the basic constituents of method rationale as defined in chapter 4. However, the connection between the goals and values of the perspective and the activities suggested in the method component is not very tightly coupled. The perspective is rather described as something that lies behind each method or method component.

**Connectivity**: The method modules must be possible to connect with each other. Each method module, viewed as part of the systems development method, should contribute to a chain of goal achievements adding to the overall goal of the specific project. Thus method modules must have the capability to express how they are connected to each other and how these connections can contribute to the overall goals linked to a specific project.

Method components are a design that intends to combine stability with flexibility with the meaning that each component could be regarded as a separate unit or building block for constructing new methods (stability) but at the same time they should be easy to model out of an already existing method and used in other contexts. An example of such a method component can be found in the problem diagram method component (Goldkuhl and Röstlinger, 1993). This component can be used in several contexts as it is a way to structure problems in order to find the root problem. Problems can exist in a business, within a system, or perhaps someone’s personal life. The problem diagram method component can be used in all these contexts and as such they present flexibility. A complete systems development method can be regarded as a container or aggregate of several method components and a method can be considered to be constituted by method components that are connected to each other. The design goal of connectivity states that the method module should be able to express how it contributes to an overall goal achievement for the systems development effort. As already stated, the method component’s perspective does contain goals and values, however not explicitly enough to be used as an interface between components in a way that would fulfil the design goal of connectivity.
Applicability: It should be possible to map the method module construct onto any existing systems development method in any situation.

The method components also meet the applicability requirement as it is described in the design goal. Nothing seems to indicate that it would not be possible to apply a method component perspective onto any existing systems development method. Even though certain aspects, such as a methods underlying perspective, might not seem to be present in a chosen systems development method it is possible to reconstruct by analysing the constituents of the perspective, i.e. definitions, categories, goals, and values.

6.3.4 Analysis of Requirements for the Starting Module Concept

My analysis of the method fragments, the method chunks, and the method components is intended to serve as a decision base for a choice of a starting module concept. The method fragments and the method chunks come from the field of method engineering which would make them ideal in an effort to redesign a method module concept for EME. The method components rather come from the field of ISD and must therefore be subjected to a higher degree of formalisation in order to be useful and accepted.

When analysing each design goal individually it becomes evident that there are certain factors that would make one module concept more suitable than another. The design goal of having self-contained method module concept has highlighted the problem of a varying level of granularity with the method fragments and the method component. This variation could possibly lead to modules that do not appear to be self-contained as they do not contain enough for them to be perceived as meaningful by themselves. The method components do have a minimum set of constituents that correspond to the diagram level of granularity (Brinkkemper et al., 1999) according to the method fragments. This property makes the method components more suitable as a starting module concept since it already has the ‘correct’ size. On the other hand, we might as well just decide to use a method fragment or method chunk as a starting module concept and simply state that it should have a size corresponding to the diagram level of granularity.

The internal consistence and coherence design goal is closely related to the level of granularity since a method module should contain enough content to be perceived as meaningful by itself. The method fragments and the method chunks do not meet this requirement for the same reason they do not meet the design goal of being self-contained. They are possible to define in such a small form that they do not have any meaning by themselves. Following the suggestions for defining a method fragment according to Brinkkemper et al. (1999) It is possible to define a method fragment as being on the concept level of granularity, having a product perspective according to the perspective dimension, and be of the
conceptual abstraction level. An example of this could be the following method fragment:

![Figure 22. Example of a method fragment]

Now, what does Figure 22 really represent or depict. Most people would say that it is a doodle depicting a man. But what does the man mean? Who drew it and why? These questions arise because the method fragment has been modelled out into a too small size to carry any meaning since it is out of context. The semantics of the symbol of the man must be interpreted in a context that would give it meaning. For example, the symbol on a plaque on a door in a hotel lobby would lead most people to think that the symbol tells us that behind that door you would be able to find the men’s toilet. The symbol on an arbitrary desk in a school classroom would indicate that someone got bored during class and started doodling on public property instead of paying attention to the teacher. However, if the symbol appeared in a diagram during a systems development project, most people involved in the project and familiar with UML (Booch et al., 1999) would recognise and interpret the symbol as an actor, a de-personalised representation of someone that might interact with the system. This shows how important it is to have a context for phenomena that are supposed to be understood and carry meaning.

The method chunks always have a corresponding process fragment for each product fragment and thus can be considered to have met the design goal of internal consistence and coherence to a larger extent than the method fragments. A combination of product and process fragments always yields a more consistent and coherent depiction of the method module. However, due to the undefined level of granularity, there is always a risk that a systems development method might be perceived as too fragmented when it is defined in method chunks that are too small to be perceived as meaningful by themselves.

In these terms the method component concept fulfils the design goal since it does not present a fragmented view of a systems development method. A method component in itself is consistent in terms of its content and presents a coherent picture of a method activity.

The *rationality* design goal states that it should be possible to understand the underlying method rationale behind each method module. Depending on the level of granularity and related to the argument above, the method fragments and the method chunks have problems with this requirement since they allow modules of different sizes that might not carry enough content to be perceived as rational. As said, this can be handled by simply deciding that they should have the diagram level of granularity. The method chunks do have some aspects that can be considered to be related to rationality in the *intentions*. These are
however not really explicitly related to the goals and values behind the suggested activities in the method chunk, but rather statements about what should be done. The method fragments do not have a rationality dimension at all. One might however, considering adding goals and values to ether concept. A fragment could be expanded by relations expressing the method rationale behind the products being produced and the activities undertaken. Similarly intentions could be re-defined as goals and values connected to the guidelines of the method chunk.

The method components have a rationality dimension in their perspective since it contains categories, definitions, goals, and values. The relation between the perspective and the method component per se is however somewhat weak and would benefit from a re-design that would yield a tighter coupling.

The *connectivity* design goal can be considered to be partly met by all three method module concepts. The method chunks are defined from a process perspective and are therefore more suitable for connectivity issues than method fragments. However, the design goal states that connection of method modules should be founded in interfaces possible to express how the method module can contribute to a chain of goal achievements. The only method module concept that has a rationality dimension is the method component concept even though it would benefit from a more explicit perspective dimension as stated above.

The applicability design goal has been met by all three method module concepts. They are possible to map onto any existing systems development method. If there are aspects in the method module concept that might not be covered by the method in itself there is always the possibility to reconstruct these aspects. Examples of this might be the perspective dimension from the method component concept which is rarely explicitly stated in any method manuals or the intentions in the method chunk concept.

6.3.4.1 Conclusion Regarding Starting Method Module Concept

Based on the analysis above, we find that we might use any of the analysed method module concepts as both method fragments and method chunks can be re-defined to always be in what we have decided to be the required level of granularity. On the other hand, the method components already have this quality, which leads to the conclusion that it would be easier to choose method components as starting module concept as they are always treated as consistent and coherent method modules. The chosen level of granularity has had strong impact on our decision process as it has consequences for multiple design goals. If a method module concept does not have the diagram level of granularity it can result in a module that can be perceived as meaningless out of context. This has repercussions on the possibilities to reach the design goals of self-contained, internal consistence and coherence, rationality, and connectivity. In other words all the design goals but applicability. This consequential problem again points in the direction of the method components.
So far the method component concept seems to be the most suitable starting method module concept. However, since the module concept is intended to be used for research in EME it needs to be formalised to a higher degree to allow manipulation in traditional method engineering activities such as modular method construction.

A fundamental point for the re-designed method module concept is its possibilities to carry and express method rationale. Even if a method fragment has the diagram level of granularity this dimension of rationality must be added. The method chunk concept’s intentions could possibly be enhanced with an explicit method rationale definition which would allow them to be used with a method rationale focus. The method component already has this property although it needs to be more formalised and explicit. Instead of regarding the goals and values in the perspective as something that lies *behind* each method component, we need to regard them as integrated parts in the method module and explain how these goals and values are related and motivate the structure and appearance of each individual method module. Thus, we must be able to create a module concept that is congruent in terms of structural issues in comparison with other method modules of similar kind. For each of the individual method modules we must also be able to define which goals and values that are connected to the proposed activities and their resulting products. In other words, the *goal relations* between the goals and the yet undefined method module parts have to be defined and understood as depicted in Figure 23.

![Method Rationale in relation to unknown method module parts](image)

In the task of re-designing an existing method concept into a method module concept suitable for EME and possible to use for research in all three method application spheres defined in Figure 14, namely the method engineering sphere, the method-in-action sphere, and the method reflection sphere. Method fragments and method chunks are already widely known in the field of method engineering whilst method components are typically not used in method engineering activities and research. Our chances of re-designing module
concepts such as the method fragments or the method chunks for EME and gain acceptance and support will be slim since these concepts already have been used extensively in various method engineering research projects. We would have a better chance of success if we choose to bring in a new concept, or at least a concept that is new in the field of method engineering, such as the method component concept. Besides this obvious obstacle, our analysis of the three method module concepts indicates that the method components are closer to our design goals than the other method module concepts. They have the ‘correct’ level of granularity from the outset which gives them a self-contained, internally consistent and congruent appearance, they have a dimension of rationality that is similar to my interpretation and definition of method rationale, and they are possible to connect to each other when their interfaces have been defined with the help of an explicitly made rationale dimension. They also meet the applicability design goal as they are possible to map onto any existing systems development method.

For these reasons, and founded in the analysis of the three method module concepts, we therefore choose the method component concept as our starting method module concept. To summarise the reason why, we can return to what I was writing about not having to re-invent the wheel. We do not need to do so. However, when someone tries to craft a wheel it is usually smarter to start with something round rather than something square.

6.4 Conceptual Re-Modelling of the Method Component Concept

In this section we will conduct a conceptual re-modelling of the existing method component concept in order to yield an improved method component concept suitable for EME. The conceptual re-modelling was conducted by a team consisting of the author and a colleague. The following sections can be regarded as the author’s interpretations of our reasoning.

One important aspect of this task was to formalise the method component concept in order to allow manipulation. We chose to formalise the re-designed method component with the help of UML (Booch et al., 1999) in order for components to be stored, retrieved and manipulated. The method component can be understood as it is depicted below in Figure 24. It describes the structure and constituents of the method component from the perspective down to the individual concepts that is applied in the method component.

The first thing one notice when analysing the method component is that is has a core consisting of three elements; procedure, notations, and concepts. They are treated as the essential part of the method component and the parts that actually carry knowledge about what activities should take place when the method component is being used. Outside the method component we find co-operation forms, frameworks, and the perspective. This is basically what we refer to when we find the method components rationality dimension to be too separated from
its prescriptive content. It is not explicit how the perspective is related to the constituents of the method components core elements.

But let us start our analysis in the core. The concepts are perhaps the most important elements in the method component concept. They must be directly related to the perspective dimension since a concept is the result of a conceptualisation. Goldkuhl’s (1994) definition of a perspective contains definitions, categories, values, and goals. Categorising different phenomena and giving them definitions is to conceptualise. The process of conceptualisation is influenced by the goals you are trying to achieve by doing so and a person usually has an explicit or implicit value base that would make him/her value one goal over another. Goals and values thus influence the process of conceptualisation. The results are of course a set of concepts that are considered useful in a certain method component. The concepts therefore have a clear relation to the perspective and can be described as “what is talked about” in the method component. The procedures are action directives that tell the user of the method component what to do. Goldkuhl (1994) describes the procedures as “which questions are asked”. There is a relation between the concepts and the procedures as the questions specified in the procedures address the concepts. Conversely, knowing the definition of a concept would enable a person to understand what questions to ask in order to find systems development phenomena that match the definition of the concept. An example of this is the concept Entity from E-R modelling (Chen, 1976). An entity is understood in loose terms as a reappearing phenomenon that would be beneficial to know about and record in a system. An entity is also possible to identify uniquely in comparison to other entities of the same kind. An entity also has certain

---

Figure 24 The original Method Component construct (Goldkuhl et al., 1998)
properties that are important to know and record. An example of an entity could be *Customer* with corresponding properties that records name, customer number, address, phone number, etc. Knowing the definition makes it easy to understand the questions that need to asked in order to specify whether a certain phenomenon can be regarded as a customer or not. You basically just have to ask questions that can answer if the phenomenon has the traits that match the definition of the concept, such as “is this reappearing phenomenon something that would be beneficial to know about and record in a system” and so forth.

Notation is described as “How the results are documented” (Goldkuhl, 1994) and corresponds to the product fragment in Brinkkemper *et al.* (1999) and Rolland and Prakash (1996). The notation is a set of rules and symbols and syntax for how results are to be captured and communicated. The relation to the concepts lies in the notation’s semantic representation of the concepts. An example can be found below.

![Figure 25 Analysis Class from the RUP. Kruchten (1999)](image-url)

The symbol above represents a analysis class from the systems development method RUP (Kruchten, 1999). The symbol is a square with three fields. The symbol itself is not a class per se, but merely a representation of something. The meaning of the symbol is contextually defined by the semantics given to the symbol by the symbol creator, in this case the persons responsible for creating the RUP. The semantics corresponds to the definition the symbol is intended to represent. In this case, the definition is “A class is a description of a set of objects that share the same responsibilities, relationships, operations, attributes, and semantics” (Rational Unified Process Version 2003.06.00.65). The definition is what is referred to when the symbol is used in context of systems development with the RUP. The relation between the notation and the concepts can be compared to Ogden and Richards’ triangle of meaning elements. Similar ideas have been put forward by C.S Peirce and Ferdinand Saussure (Fiske, 1994). Thus the relation between the notation and the concepts is strong, evident and clear. The relation between the notation and the procedures is dependent on knowing the concepts. Of course, knowing what types of questions to ask means that the method user knows what concepts s/he is trying to find and s/he also probably know what symbols s/he should use to represent his/her findings with in a given systems development context. However, in my re-designed method component concept I choose to refer to procedures as *actions* as we find this term to be more fitting.
This conceptual analysis of the original method component concept shows that there is a strong connection between the process and the product elements. Since we are trying to create a method component concept that can be considered as self-contained and internally consistent and coherent in order to have method components that have an ability to carry meaning by themselves it is important that we keep this strong bond in the core of our re-designed method component concept. However, since we have a specified level of granularity to take into consideration this aspect must be added. Method professionals intuitively refer to the artefacts when they are talking about method modules. Furthermore, since the artefacts are identifiable objects by themselves and also a container of the results in the notation used for creating the artefact, the artefact is added to our first formalised version of the re-designed method component concept. The formalisation is done with the help of a class diagram from UML (Booch et al., 1999).

As artefacts often are used to create other artefacts we add two associations that have the capability to express how, for instance a use case diagram can be considered to function as input for the activity of producing a use case model in the RUP.

So far we have formalised the core of the method component concept with an addition an artefact element and associations for artefact reuse. The result can be seen in Figure 26.

Another problem with the original method component concept is that even though actions are specified there are no possibilities to specify any actors. This raises problems since a lot of contemporary systems development methods have specific actor roles, i.e. roles that people involved in the systems development project take when performing certain activities. An example of this is the RUP where there are several actor roles such as system analyst, test manager, or software architect (Kruchten, 1999). Even the new agile methods have specified...
roles such as Scrum’s scrum master, or product owner (Schwaber and Beedle, 2002) or extreme programming’s coach, programmer, or tracker (Beck, 2000). This indicates that it would be advisable to include actor roles in the core of the re-designed method component concept.

Another aspect that becomes apparent is the need to formalise the internal structure into separated method elements. By doing so, we can define the re-designed method component concept to have a defined set of constituents. This is done through a generalisation association between each individual method element (concept, notation, artefact, action, and actor role) and the container class method element. A container class is a class that does not have any instantiations by itself, only instantiations in the form of other classes it is related to (Booch et al., 1999). A result from this is that we can adhere to the design goal of internal consistence and coherence. A method component is considered to consist of method elements, shown above in Figure 27 which together leaves the method component with a homogenous appearance that would enable future method component congruence. The result is a method component concept with is able to function as a self-contained unit with a possibility to be meaningful by its own accord and thus partly fulfil the design goal of self-contained. The internal structure of the method component can be understood as an aggregate of defined method elements which together becomes the core of the internal structure of the method component. This is represented.
though the use of an aggregation symbol in the formalised description of our re-designed method component concept in Figure 27.

Another aspect that must be handled is to make the relation between the method components underlying perspective explicit in relation to its individual method elements. Goldkuhl et al. (1998) describe the perspective to constituted of categories, definitions, goals and values. However, since the categories and definitions are ways to interpret and denote specific concepts there is no apparent need for formalisation of these parts in our UML description. The concepts are already there and the categories and definitions are parts of the conceptualisation. These two constituents of the perspective do exist but can be considered to be already present in the core of the method component, although indirectly. Furthermore, we have not indication that an addition of the categories and definitions would be useful in EME so far. Thus, we leave them out.

The only constituents left in the perspective are the goals and the values. These are the fundamental parts of method rationale as defined in chapter 4 and necessary for the re-designed method component concepts ability to be regarded as truly self-contained. In chapter 4, method rationale is defined as goal relations between the method description and the goals (goal rationale) and a relation between the goals and the values the goals are anchored in (value rationale). A modularisation of a systems development method into method components would allow for these relations to be explicitly formulated in relation to each of the method elements that constitute the core of the method component. I have already argued for how the primary parts of the core (concepts, actions, and notation) are related to the perspective and the additions I have made since are related to the perspective in a similar fashion. An actor role is also a result of a goal oriented and value based categorisation which results in a conceptualisation which is defined. For example the actor role customer is considered important in extreme programming (Beck, 2000) as a customer is defined as:

“The XP Customer role has the responsibility of defining what is the right product to build, determining the order in which features will be built, and making sure the product actually works.”(Beck, 2000)

As a result, the actor method element also has clear relations to the perspective’s goals and values.

The artefact method element is considered to be a record of the results expressed through notation and contains various symbols or textual descriptions that correspond to the concepts (what the document is about) and the actions taken for in the production of the document (the questions asked). Furthermore, the artefact can be regarded as a container of other artefacts. By this, I refer to documents in a very loose sense. An example of a document in this sense is
something that is recorded during systems development projects in order to be communicated in some way. Thus finding a single analysis class would have to be considered a method component because finding classes is considered to be separated activities as they result in singular artefacts that are used to create new artefacts. In the RUP individual classes are recorded in diagrams which are later on used to analysis class diagrams. Since, there is nothing that necessarily says that all classes have to be used in every analysis class diagram they have to be recorded separately and be treated as individual method components residing on the diagram level of granularity since even a single analysis class can be considered to be an individual artefact. Artefacts that can be used instantly or later on in the overall systems development process for creation of other artefacts, such as analysis class diagrams or database design (Booch et al., 1999). This would mean that the symbol in Figure 25 can be considered as an artefact from the analysis class method component. Thus, every artefact has a defined purpose for its existence and an underlying method rationale that explains why the artefact is produced.

![Diagram](image-url)

**Figure 28** The Method Component Concept v 1.0 (The Internal View)

Hence, it is possible to treat all method elements in the same way in relation to the perspectives goals and values. As a result we add an association describing the goal rational relation to the method elements and a value rationale association to describe the relation between goals and values. We also add the goal achievement and value anchoring associations. We have chosen to omit the goal and value conflict associations in the formalised re-designed method component concept. This does not mean that we deny that they exist; it only
means that it is not necessarily so that these types of conflicts will be found and need to be understood. As a matter of fact, we have yet to find one in a contemporary systems development method. However, one must consider the possibility that a conflict of this type may very well exist.

Finally, we add an association between the goal and the aggregated method component describing the overall goal of the method component. The overall goal is the primary goal a method user would reach when using a given method component. An example of this is the Business Use Case Model from the RUP. Its overall goal can be defined as “To agree upon the business context in order to plan and follow up subsequent software development” (Kruchten, 1999). Figure 28 depicts the re-designed method component concept. As such, it represents the internal view of how a method component can be defined and understood.

6.4.1 External View of a Method Component
So far we have defined a method component’s internal structure and made the relation between these structures and the underlying perspective explicit. By doing this we have fulfilled the design goals; self-contained, internal consistence and coherence, and rationality. However, we have yet to handle the connectivity design goal which states:

Connectivity: The method modules must be possible to connect with each other. Each method module, viewed as part of the systems development method, should contribute to a chain of goal achievements adding to the overall goal of the specific project. Thus method modules must have the capability to express how they are connected to each other and how these connections can contribute to the overall goals linked to a specific project.

In order to fulfil this design goal we let the overall goal of each method component contribute to a chain of goal achievements. This connects the method components to each other through the help of method rationale. Figure 29 illustrates how the re-designed method component construct should be perceived when considering them being parts of a complete systems development method and hence responding to the criterion of connectivity. Through the external view we focus on how a specific method component contributes to a chain of goal achievements, sharing its rationale through an interface and adding to the overall goal of a systems development project. This depiction is again formalised with the help of UML (Booch et al., 1999). A difference compared to the internal view of the method component is that it is more likely to find goal contradictions on this level. It may be the case that a method engineer is trying to reconstruct an informal or in-house development method which happen to have these types of contradictions. One might also consider two method components that fulfil exactly the same task as contradicting since they might
force a method user to perform essentially the same task twice. Fulfilling the same goal twice can hardly be considered as positive in relation to the overall goal of the development project in most cases. An example of this can be found in UML where the *Sequence diagram* and the *Collaboration diagram* have the same end result, namely to clarify the flow of order objects interact within a system.

6.4.2 The Interface and Content of a Method Component

In order to address the relevant aspects in the connectivity design goal where method components should be connected to each other with the help of method rationale we must define how this is done in more detail. Just following UML will not do since an association can be arbitrarily defined between any two classes. Hence we must define an interface for the method component concept that can describe a chain of goal achievements. When we analyse what a method component offers in terms of goals they fulfil, we acknowledge that the offered method rationale is expressed in the results recorded in the method component’s output artefact.

Furthermore, the method component’s requirements can be expressed through its input artefacts. Thus we choose to define a method component’s interface as the method component’s overall goal and artefacts. The latter are classified as prerequisites or deliverables. Our use of the term prerequisite should not be interpreted as strict prerequisites that are a must. We treat methods as heuristic procedures or heurithms (Langefors, 1995) and consequently prerequisites should be treated as recommended inputs.

6.5 Example of a Method Component

In the following sections we will present an example of a method component. The example will not be formalised with the help of UML for the sake of readability. Though, it is important for the reader to know that the presented method component can be formalised and described in UML for the sake of manipulation. It is also the case that it might not always be necessary to have a full description of the entire method component. An example of such a situation can be when a person responsible for a method implementation project has to

![Diagram of the Method Component Concept (The External View)](image-url)
decide which method components to implement in an organisation, much like chapter 5.

We have focused on creating a Business use case model component and one can argue that the choice is somewhat arbitrary. However, the importance lies in that this example is performed on a rigorous systems development method, which means that the method elements often are intertwined with each other and have to be separated for clarity. There is a clear difference between describing rigorous systems development methods such as the RUP (Kruchten, 1999), compared to an agile method such as extreme programming (Beck, 2000).

The presentation of the Business use case model component is structured into two sections. In the first section we discuss the method component’s content, or internal view, with a focus on the actions it contains in terms of activities specified in the original method description. In the second section we present the method component’s interface and how it the method rationale it offers should be perceived.

Furthermore, we illustrate the external view through an additional example where the connection between the Business use case model component and a Business use case component is addressed. This is done to show how the method components can be connected together as building blocks in order to create an aggregate that can be considered a complete systems development method. The Business use case component is only illustrated through its interface as this is sufficient for this example.

6.5.1 The Business Use Case Model Component Content

The description of the Business use case model is presented with a focus on the actions that are prescribed in the content. The content is what a method user would see if s/he applied an internal view on the method component. The content contains method elements as defined in section 6.4. The fundamental method elements are the notation, concepts, and the actions needed to accomplish the artefact that is the deliverable of the method component. Furthermore we find the actor roles that carry out these actions in this part of the method component description.

For the sake of the example it will not be necessary to list all concepts and all notational conventions that are defined and suggested. The notational issues are all solved with the help of UML (Booch et al., 1999) and the concepts are present through semantics of the symbols and the syntactic rules that UML defines, much in the same way as described during our conceptual re-design of the original method component concept¹. Thus, a description of these method elements will be omitted.

¹ The description of the relation between concepts, notation, and actions are elaborated and analysed in the discussion leading to our re-design of the Method component concept into v 0.1 (see Figure 26).
Other aspects that will not be covered in the example method component are the individual goals and values that motivate each individual method element. Even though these goals and values must exist for conceptual reasons, there is not much that indicates that every method component description really needs to be described in that much detail. This can be considered a trade-off between precision and cost. The decision of implementing a specific method component in a development organisation is rather based on the goals and values that motivate each method components’ main deliverable, the primary artefact. Since the artefact contain other atomic method elements, such as individual concepts, these concepts and their goals and values can be said to be present in the primary artefact. Nothing points in the direction that it would be useful to model the goal achievement and value anchoring structure for each method element in practical method usage situations. Hence, an analysis of these aspects will also be omitted.

In our Business use case model component, we find two actions in the RUP that have the Business use case model artefact as a deliverable. These two actions are Find business actors and use cases and Structure the business use-case model. When working with a rigorous systems development method such as the RUP the self-contained criterion is non-trivial to apply. The two actions we discuss have several artefacts as deliverables, and that is not in line with our specification of a method component having only one. This problem can be handled by letting steps and actions reoccur in other method components. A description of how this is done will follow the main presentation of the Business use case model artefact.

Until then, we will stick to a description of the actions that are related to the method component’s main deliverable, namely, the above mentioned Business use case model artefact.

Table 2 illustrates the result from our analysis. The table is split into two columns; the left contains actions gathered from the RUP, and the right contains the chosen steps that are related to the method component’s deliverable. The action Find business actors and use cases have nine steps in the original method description. From this set, we have chosen a smaller set of five steps that are suggested as necessary in the RUP to build a business use case model and a sixth step where a business use case model is evaluated. Therefore, this subset is based on the goal rationale of these actions and whether or not the goals of these actions contribute to the goals of the output artefact that is the overall goal of the method component.
Table 2. Method Component Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find Business Actors and Use Cases</td>
<td>Develop an Outline of the Workflow of Business Use Cases&lt;br&gt;Describe How Business Actors and Use Cases Interact&lt;br&gt;Package Business Use-Cases and Actors&lt;br&gt;Present the Business Use-Case Model in Use-Case Diagrams&lt;br&gt;Develop a Survey of the Business Use-Case Model&lt;br&gt;Evaluate Your Results</td>
</tr>
<tr>
<td>Structure the Business Use-Case Model</td>
<td>Establish Include-Relationships Between Business Use Cases&lt;br&gt;Establish Extend-Relationships Between Business Use Cases&lt;br&gt;Establish Generalisations Between Business Use Cases&lt;br&gt;Establish Generalisations Between Business Actors&lt;br&gt;Evaluate Your Results</td>
</tr>
</tbody>
</table>

The second action, *Structure the business use-case model*, contains five steps, which is equal to the number of steps prescribed in the RUP. These steps are not atomic with regard to their deliverables, since their results are *reworked* artefacts of the business use case model, the business actors and the business use cases. The solution is to let these steps *reoccur* in the consequential *Business actor component* and the *Business use case component*. Accordingly, these steps can be considered as being parts of *all three* components, otherwise it is not possible to consider them independent. This is a trade-off between the redundancy problem and the problem of a self-contained method component. This also solves the problem of having multiple deliverables as described on the previous page. By letting each method component have only one artefact defined as its deliverable we can ensure that the offered method rationale of each method component will be possible to identify. At the same time, as we let required inputs such as business actors and business use cases reoccur we can ensure that the business use case model can be considered to fulfil the self-contained design goal.

Finally, we have to consider the actor role associated with this method component. According to the RUP the *Business-process analyst* is responsible for the Business use case model and should therefore be included in the method component. Furthermore, we can identify two stakeholders, *customer* and *end user* that are important during work with the method component’s outcome. Consequently, they are included in the Business use case model component’s content part as actors.
6.5.2 The Business Use Case Model Component Interface

So far we have only addressed the internal view of the Business method component. However, in order to connect method components to each other we need to specify an interface that has a possibility to express the method rationale offered by the method component. The main deliverable from each method component is the artefact and the goals it fulfils along with the values that motivate why the goals can be considered important. Exactly how a task is executed is not interesting for an external viewer of a component when s/he is executing tasks such as the ones described in the method implementation project in chapter 5. In such situations, the user of a method component is primarily interested in the results offered by the component (the deliverable) and the required inputs (prerequisites) needed to achieve these results. Thus, practical tasks concerning multiple method components only require the method user to apply the external view of the method components s/he is working with.

As stated in 6.4.2 a method component’s interface is constituted by the overall goal of the method component and the related artefacts that are of the type prerequisite or deliverable. The overall goal is communicated to adjacent method components through the deliverable as the deliverable can be considered to express the goal it has fulfilled through its notation. An example of this could be the goal of the Design model in the RUP which can be defined as “Creating an object model capable of describing the realisation of use cases, and serving as an abstraction of the implementation model and its source code” (Kruchten, 1999). By executing the Design model method component a method user would have an artefact that fulfilled that particular goal.

Thus, the point of departure for our example is the method component’s deliverable. Only one artefact is defined as deliverable in order to align with the chosen diagram level of granularity. This also makes the method component capable of expressing its goal through one single artefact. In order to create this deliverable the Business use case model component has six artefacts defined as its prerequisites: Business modelling guidelines, Stakeholder requests, Glossary, Vision, Business actor, and Business use case.

The list in Table 3 describes the artefacts that are used in connection with the Business use case model. It is structured in two columns where the first column contains the artefacts name and the second column contains the artefacts’ different roles in relation to the method component. The artefact roles describes if each of the related artefacts should be considered to be a deliverable or a prerequisite.
Table 3. Method Component Interface: Artefact List

<table>
<thead>
<tr>
<th>Artefact</th>
<th>Artefact Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Use Case Model</td>
<td>Deliverable</td>
</tr>
<tr>
<td>Business Modelling Guidelines</td>
<td>Prerequisite</td>
</tr>
<tr>
<td>Stakeholder Requests</td>
<td>Prerequisite</td>
</tr>
<tr>
<td>Glossary</td>
<td>Prerequisite</td>
</tr>
<tr>
<td>Vision</td>
<td>Prerequisite</td>
</tr>
<tr>
<td>Business Actor</td>
<td>Prerequisite</td>
</tr>
<tr>
<td>Business Use Case</td>
<td>Prerequisite</td>
</tr>
</tbody>
</table>

The method component’s total method rationale can be expressed through the range of goals specified in the method component. Goals are related to each other in complex structures and can be identified or reconstructed for each method element if needed. As already stated, we will not conduct an analysis of the complete content of the method component’s method elements and their foundation in goals. In this example of a method component’s interface we choose to focus on the goals related to the deliverable as these can be considered to be the most relevant as they present the offered method rationale explicitly. Table 4, below, also contains two columns; the left column contains the identification numbers of the goals, and the right column contains the goal descriptions themselves. All goal descriptions are described in plain text to help a method user to understand the goals and have an easier task of appreciating whether a certain offered method rationale can be considered as important or not. Often, the goals and values of a method component’s artefact are not explicitly stated in a method description. In such situations, a modularisation of the systems development method into method components needs to involve an analysis of, at least, the method rationale offered by the primary deliverable. Otherwise it would not be possible to specify an interface for the method components and it would be impossible to adhere to the connectivity requirements set in our design goals.

Table 4. Method Component Interface: Goal List

<table>
<thead>
<tr>
<th>Goal Number</th>
<th>Goal Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0) Overall</td>
<td>To agree upon the business context in order to identify architecturally significant behaviour that should be automated and plan and follow up subsequent systems development</td>
</tr>
<tr>
<td>1</td>
<td>To agree that you understand the business context before you continue developing the system</td>
</tr>
<tr>
<td>2</td>
<td>To identify architecturally significant behaviour that should be automated</td>
</tr>
<tr>
<td>3</td>
<td>To plan and follow up on the business modelling effort and subsequent systems development</td>
</tr>
</tbody>
</table>
We have identified three goals for the resulting artefact of the Business use case model component by tracing the goal rationale for the artefact method element. Furthermore, we have summarised these goals into an overall goal that is used for the method component in order to be able to express an overall goal for the Business use case model component. The business use case model is used to understand the business context for the future information system, to identify which tasks in the business context are appropriate to automate, and essential for planning and following up actions in the project. All these aspects are reformulated into an overall goal that can be defined in the Business use case models interface.

Although an analysis of the value anchoring structure is not necessary in order for a method user to understand the overall method rationale for the Business use case model component in a practical method application situation we might as well show what such an analysis would look like. If finding explicitly stated goals can be difficult, it is even more seldom that a quick overview of a systems development method reveals the underlying values the goals are anchored in. Rather, the value rationale dimension is taken for granted or left in the hands of the method user to explore. One can compare this to a situation where a method user asks himself the question if it is important to reach the goal of a certain method component, thus appraising or making an evaluation of the contribution of method component’s goal. In a situation where a method is implemented, such as in chapter 5, a professional method implemenenter rarely has to consider the underlying values as these are either obvious or tacitly understood. Nevertheless, in this example we have conducted an analysis of the underlying value anchoring structure that constitutes the value rationale for the Business use case model method component. By doing so, we can present an even more coherent example of our exemplifying method component.

An analysis of the Business use case model method component’s value anchoring structure is presented in Figure 30. The figure is easiest understood when studied from the bottom and up since the goal is situated there. Above the goal, values are described and presented hierarchically, meaning that values presented are considered anchored in the value directly above.

The value anchoring structure is limited to the overall goal (0) of the method component presented in Table 4 since this is the only goal visible in the method components interface. The analysed goal has a relation to a value in accordance with the definition of method rationale. This is the value rationale as it is described in section 4.3. Recording the primary value for a method component is basically a process where the goal statement is re-formulated into a statement that stresses the importance of reaching the defined overall goal. This value must then be analysed in order to ascertain what other values that might be related to this primary value and how these are hierarchically structured. The top value can be considered to be fundamental values that correspond to ethical and moral standpoints. Of course, there is a possibility to find other value structures
that might be completely different. The overall goal of something might be separated from moral aspects at first glance. It is often the case that the usage situation decides which moral and ethical standpoints are indirectly enacted. A knife with its overall goal of functioning as an efficient instrument for cutting can be used in many ways. However, the value anchoring structure that motivates why this goal is desirable can have many forms. The goal “excellent cutting property” might be used in situations where a knife carrying user is constructing something, preparing food, or perhaps as a weapon. It is basically the usage situation that decides the value anchoring structure and this tells us that this type of analysis of every method component does not need to be done. However, one must be aware of the fact that there is an underlying value anchoring structure that motivates each and every existing method component. Otherwise, the method component would not have been defined in the first place. This also tells us that there are possibilities to find new application areas for method components if a method user decides to define the value rationale relation differently compared to the original method constructor.

![Value anchoring structure for the Business Use Case Model Method Component](image)

**Figure 30.** Value anchoring structure for the Business Use Case Model Method Component

### 6.5.3 Connection of Method Components

Having defined the interface for our Business use case model method component, we now have a possibility to connect it to another method component. One of the prerequisites defined in the interface for our example method component is the Business use case. Hence, these two components’ relations to each other have to be defined in our example. We have restricted the
example to only cover one instance of goal achievement to show how the connectivity design goal is met.

When a method user is connecting method components s/he is applying the external view of the method component concept. By applying an external perspective on method components, the method user is only paying attention to the interfaces of the method components s/he is studying. The content of the method component is considered to be hidden inside the internal view.

In Figure 31 two method components are illustrated, each one exposing their interfaces. To the left we find the Business use case component and to the right we find our exemplifying Business use case model component. Our example method component has the deliverable from the Business use case method component as a defined prerequisite. This example of method component connection is restricted to only one prerequisite. In fact, the Business use case model method component has 5 additional artefacts defined as prerequisites as shown in Table 3. Furthermore, to complete the interface we present the overall goal of each component. The overall goal of the Business use case component can be defined as “To describe the business process value-added aspect” the overall goal of our example method component is the same as defined in Table 4 “To agree upon the business context in order to identify architecturally significant behaviour that should be automated and plan and follow up subsequent systems development”. However, for the task of connecting just these two components, only the goal of the Business use case method component is required.

![Figure 31 Connection of two Method Components](image)

In Figure 31 we show connection between our two components. The connection is represented in an informal fashion to support readability. However, since the method components have a formalised representation also it is easy to formalise the connections between method components if needed. This can be achieved by applying UML definitions in line with how they have been applied in Figure 29.
6.6 Chapter Summary
This chapter has covered three activities that intended to define a method module concept suitable for EME research. When placed in my synthesised framework for EME in Figure 14 the research presented in this chapter resides in the method engineering sphere. The chapter started with a return to the requirements for a method module notion set in chapter 4. These requirements were analysed in order to elicit design goals for the new method module concept. We ended up with the following design goals:

- **Self-contained:** It must be possible to treat the method module as a self-contained part of a systems development method with regard to the guidelines that describe the deliverable and the process of producing such a deliverable. This way a method module can be modelled out from a systems development method or act as a building block when constructing new methods or during method integration.

- **Internal consistence and coherence:** The method module must be perceivable as an internally consistent and coherent entity. Another way to put it is to say that the construct should be stable over time and without lose ends. As a result the method module will be perceivable as meaningful. The principle is an ambition to create congruence amongst method modules and homogeneity in systems development methods.

- **Rationality:** Actions are performed and prescribed for reasons. Thus a method module must have an identified target state, a purpose for its existence. The arguments are motivated by the modules inherent method rationale and address the goals and values that the method module realises.

- **Connectivity:** The method modules must be possible to connect with each other. Each method module, viewed as part of the systems development method, should contribute to a chain of goal achievements adding to the overall goal of the specific project. Thus method modules must have the capability to express how they are connected to each other and how these connections can contribute to the overall goals linked to a specific project.

- **Applicability:** It should be possible to map the method module construct onto any existing systems development method in any situation.

Having the design goals defined gave us a possibility to perform our second activity of choosing a starting method module concept for our re-design process. We compared the design goals to three method module concepts previously
introduced in chapter 3; Method fragments (Brinkkemper et al., 1999), Method chunks (Rolland et al., 1999), and Method components (Goldkuhl et al., 1998). The analysis of the three module concepts in combination with our decided level of granularity from chapter 5 resulted in a choice of the method component concept as our starting point for our re-design activity.

The third, and final, activity covered in this chapter was a conceptual analysis of the method component concept in an attempt to re-design it to meet the chosen design goals. During this activity we also formalised our re-designed method component concept with the help of UML (Booch et al., 1999). The formalisation process and its results allow the method component to be stored and manipulated if needed in various instances of EME. An example of how the method component concept has been applied in this fashion can be found in Karlsson (2005). The resulting formalised version of the re-designed method component concept and its two views (the internal view and the external view, respectively) is presented in Figure 28 and Figure 29. Along with UML descriptions of the re-designed method component we also define the interface of the method component as containing the input artefacts (called prerequisites), the output artefacts (called deliverables), together with the deliverable artefact’s overall goal.

The chapter ends with an example where we present a method component according to our re-designed version to show how a method component should be understood. We chose to exemplify our re-designed method component concept by showing the internal view of the Business use case model component. After that, we presented an external view of our method component and showed how it can be connected to another component. In our example we connected our method component to the Business use case component.
7. Application of Method Components in Method Reconstruction

Chapter 5 ended with some conclusions regarding how the project assessment phase was conducted in the E-Gov project. A main critique concerned the chosen approach for method reconstruction in order to find areas for improvement. The reconstruction phase was oriented towards problems in the existing development process and followed by an attempt to find support from the RUP for tasks that the consultants found necessary. The reconstruction phase and subsequent task of finding support in the RUP followed the reasoning below:

1. What are the problems in the current development situation?
2. What type of support could handle the problems?
3. What in the RUP can provide the desired support?

We received indications that this approach for method reconstruction might yield the same results regardless of what the consultants might find. A major reason for this was a statement given by one of the consultants where she explained that the chosen RUP disciplines (Requirements, Configuration and Change Management, and Test) were typically suitable for initial RUP implementation as these aspects are often missing from a RUP perspective. A problem with this view is that it is likely that consultants working with these premises might be biased and only did what they usually do.

We got the opportunity to conduct the method reconstruction task in an assessment of another potential project, the PSU (Produktionssystem Utrikes) project. This let us employ the method component concept and the method rationale definition in a practical setting and explore the communicative power of method components in the task of reconstruction of informal development methods in a case study. All the work described in this chapter was conducted by the author and the same colleague as in chapter 6.

7.1 Chapter Overview

This chapter is devoted to two activities. The first activity enabled us to test our re-designed version of the method component concept in an attempt to reconstruct an informal development process. In chapter 5, during the project assessment of the E-Gov project we had noticed some weaknesses in the current approach for method reconstruction and we got the opportunity to test alternative approach for method reconstruction based on method components and method rationale. The input for this activity was the newly re-designed method component concept and the output consisted of our elicited reconstruction results. The activity is described in section 7.2.
The second activity is an evaluation activity where the re-designed method component concept once again is conceptually analysed in order to find new requirements. This activity also has the consequence that the method component concept was allowed to be re-designed as indicated by the arrows going both ways. The same circumstances were applicable in the design goals and the initial method rationale definition. Any changes in definitions and design goals in chapter 7 are founded in the results from the reconstruction activity. A figure depicting the details of chapter 7 is found below.

![Diagram](image)

**Figure 32** Chapter 7 details

### 7.2 Case Setting
The PSU (Produktionssystem Utrikes) was a project that aimed to adapt the Postal services post systems and workflows to new international directives stated by the Universal Postal Union, the United Nations’ agency responsible for setting rules for international mail-exchanges and the primary forum for cooperation between postal services in 191 member countries. This is considered a highly business critical project with respect to the production systems. In the event of a failure that would result in a halt, the cost is estimated to be running at approximately USD 60,000/hour. Thus, it is understandable that Posten IT would want to have well defined processes for the continued development and maintenance of their systems.

At the time of the case study the PSU project already had delivered part of the systems and they were up and running, although not completely. The project could be characterised as a large project with a time frame surpassing one and a half years. Overall, three major deliveries was planned.

#### 7.2.1 The PSU Project Assessment
The project assessment of the PSU project also focused on the aspects described in chapter 5, such as General method knowledge, RUP knowledge, Attitude towards RUP, and Current tasks and method support. However, most of these aspects are beyond the scope of this chapter. Instead this chapter will focus on the last aspect; Current tasks and method support, but conducted differently compared to the E-Gov project. Below is a summary of the results from the
project assessment of the PSU project along with the consultants’ decision regarding possible RUP implementation.

The consultants collected relevant information about the development team and tried to understand the problems and strengths of the informal development process they applied. The team in the PSU project applied a development method conceived by themselves. The informal development method was put together by the persons involved and was a mix of method components the developers had previous experience from. All in all, the consultants considered the development process to be functioning well as far as they could tell. The only major problem was that the knowledge of how the development process was put together was completely implicit. Not even the persons involved felt that they had a complete picture as they were often only concentrating on parts of the process and lacked an overview. Still, the project seemed to perform efficiently even though they had not implemented ATUP yet.

The consultants came to the conclusion that the PSU project would not be suitable as a pilot project. As far as they could tell the team seemed to have an idea of their own that was similar to the RUP and the differences would probably be something that could be ironed out when the RUP implementation finally would come to the PSU project.

The consultants found, although not expressed in those specific terms, a development team that apparently was able to achieve rationality resonance with the help of their own implicit development method. As far as judging whether the team needed public rationality for specific tasks or not, they had chosen to rely on their own private rationalities and found support in method components they had previous experience with. All in all, the PSU project had a rational development process but it would be difficult for a new team member to internalise the method rationale of the process since it was completely implicit and informal. Implicit in this aspect refers to the fact that the development method was not recorded anywhere and informal means that there were no prescriptive conventions defined for how project results would be communicated within the project. This means that while people involved in the development process appreciated input in the form of artefacts produced by someone else were in no position to demand such artefacts since no such rules were defined explicitly. It would also be difficult to use the development process conceived by the PSU project team as a success story as they initially planned if it was not possible to communicate. The consultants thought that it would be a good idea to inspire other teams by communicating a rational and well functioning development team but it would be difficult since the process was not properly defined and described in way that would permit feasible knowledge transfer.
7.2 Method Reconstruction using Method Components

The need for an explicit method description gave us the opportunity to apply the viability of using the method component concept in a method reconstruction task. Our ambition was to explore the method component as a communication concept in such a task. As shown in previous sections, a project assessment phase during a RUP implementation covers a multitude of aspects ranging from general project data to a description of how the project currently conducts systems development. This latter assessment aspect is typically covered and explored by interviewing the persons involved in the project. They describe how they work and the consultants identify problems. The problems are then considered as something that must be solved with the help of public rationality found in the RUP. Finally, appropriate selections from the RUP are chosen and implemented in the project to solve the problems.

Bringing in the method component concept to the method reconstruction phase would allow a different approach that could give additional depth to project assessments. Initially the proposed approach for the project assessment phase yielded a lot of results that the consultants found useful for planning the future method implementation. However, the consultants’ assessment did not cover an approach for how an implicit development process should be reconstructed and how the results from such an endeavour could be recorded. Applying a method component concept also keeps method rationale in the foreground since it is what gives the method components meaning and connects them to each other through their interfaces, containing artefacts of the type prerequisite or deliverable, along with the overall goals of the method components expressed in the result from the deliverable artefact. The process of reconstructing the implicit systems development in the PSU project had the following general structure:

1. Apply a method component perspective on the implicit process.
2. Ascertain what method rationale explains why they are doing things the way they are.
3. Compare the process to the method rationale offered by the RUP.
4. If needed, make adjustments by adding or removing method components.

The steps above involved a series of activities that explains what a person is supposed to do during method reconstruction with the help of method components.

Step 1 "Apply a method component perspective on the implicit process" means that the person responsible for the method reconstruction applies a method component perspective on the implicit development process by listing all artefacts that are produced within the project. Since the method components’ level of granularity is defined to correspond to singular deliverable artefact a list...
of all artefacts would in fact be a list of all method components. At this time it is not necessary to find specific artefacts in existing methods as there might very well be the case that there are no such components to be found. If an artefact is elicited it is defined as a deliverable from a method component with the same name. No relations between specific method components are defined.

Step 2 “Ascertain what method rationale explains why they are doing things the way they do” is carried through by an analysis that shows how the method components are related to each other. This means that the person responsible for the reconstruction considers and defines the interface of each method component until each method component relation is understood and the overall goal achievement structure is defined. The task of connecting method components to each other is done by analysing the artefacts role to other method components by asking which artefacts can be considered to be either prerequisites or deliverables to which other method components. The individual goals of each method component are understood as the goals reached by producing the deliverable artefact. Having defined a goal relational structure between the components enables the person responsible for the reconstruction process to communicate the development method to other people. Note that we do not say that it is always possible to elicit a full chain of goal achievements defined as method components in every implicit or informal development process. An analysis might very well lead to the conclusion that there are missing pieces in the goal achievement structure as some aspects of systems development might be omitted completely. However, this is not necessarily something negative as a reconstruction analysis with a method components perspective would clarify which parts of the development process that might need method support the most. This process of reconstruction is of course conducted iteratively together with representatives from the project team.

The project assessment of the PSU project had some additional aspects that had to be covered since the work at Posten IT ultimately aimed to implement the RUP. This means that we had to conduct an analysis of how the reconstructed development method compared to the RUP. Hence, two additional steps were added.

Step 3 “Compare the process to the method rationale offered by the RUP” meant that we compared the reconstructed systems development method with RUP in terms of how method components helped to contribute to the overall goal of the development project. During this step it was possible to define discrepancies between how method components where used and connected. It was also possible to better understand the role of each method component as their goal relations became more clear when compared with a public rationality offered through the RUP.

Step 4 “If needed, Make adjustments by adding or removing method components” is intended to let the persons responsible for the reconstruction process to adjust the reconstructed systems development into a set standard. In
our case, the consultants asked themselves whether they should enforce a RUP perspective on the PSU team. This would possibly require that they would have to make some alterations in their systems development method in order to address probable conformity issues. In our reconstruction case study, we never performed step 4 as the consultants deemed it unnecessary to enforce a RUP perspective on an otherwise well functioning project team.

Compared to the approach for method reconstruction in the E-Gov project assessment this approach yields a description of the development process that can be communicated to others more easily since it would result in a diagram with the possibilities to give an overview of the development method. The diagram would consist of method components that were connected to each other through goal rationale showing how the sum of method components together contributed to the overall goal of the development process.

7.2.1 Method Reconstruction in the PSU Project

The application of a method component concept perspective in the PSU project was conceived by following the steps suggested above. We had in total three meetings with two key persons in the PSU project’s team involving the project leader and one other developer.

7.2.1.1 Step 1

The team members were asked what they produced in terms of artefacts during development. This was important since the artefact is considered the primary deliverable for each method. Our analysis from Step 1 showed that they indeed had adopted a lot of method components that could be found in the RUP. Most of the developers had experience with UML so using different method components such as Use case diagrams and Design classes felt natural for them. As a consequence, much of our initial analysis from Step 1 suggested that PSU already had adopted the RUP as a systems development method. However, there were some differences between what is suggested in the RUP and what was taking place in the PSU development processes.

It also became evident that parts of their systems development differently compared with the RUP. This was clear when we recorded the use of the method components Test case template, and the List of measures. These method components do not exist in the RUP. They are rather adapted variants of method components conceived by the PSU team and founded in their previous experience from testing software. At that time during the analysis we did not give much attention to the content and overall goals for these method components. We just recorded them in order to get a clear overview of the sum of artefacts produced, and consequently, which method components that could be considered part of the PSU systems development method.
The results from Step 1 are shown in Table 5 below.

<table>
<thead>
<tr>
<th>Artefact List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Process Description</td>
</tr>
<tr>
<td>2 Glossary</td>
</tr>
<tr>
<td>3 Supplementary Specifications</td>
</tr>
<tr>
<td>4 Use Case Specification</td>
</tr>
<tr>
<td>5 Business Rules</td>
</tr>
<tr>
<td>6 Activity Diagram</td>
</tr>
<tr>
<td>7 Use Case Model</td>
</tr>
<tr>
<td>8 Actor</td>
</tr>
<tr>
<td>9 Use Case Diagram</td>
</tr>
<tr>
<td>10 User Interface Prototype</td>
</tr>
<tr>
<td>11 Analysis Class</td>
</tr>
<tr>
<td>12 Analysis Class Diagram</td>
</tr>
<tr>
<td>13 Design Class</td>
</tr>
<tr>
<td>14 Database</td>
</tr>
<tr>
<td>15 Component</td>
</tr>
<tr>
<td>16 Build</td>
</tr>
<tr>
<td>17 Test Case Template</td>
</tr>
<tr>
<td>18 Test Case</td>
</tr>
<tr>
<td>19 Test Data</td>
</tr>
<tr>
<td>20 List of Measures</td>
</tr>
<tr>
<td>21 Test Log</td>
</tr>
</tbody>
</table>

7.2.1.2 Step 2

After we got a clear picture of what artefacts they produced we focused on why they produced them in Step 2. For example, a question concerning the artefact Actor focuses on where else in the development process the Actor artefact is required as input. The answer they gave us was that the Actor was needed when producing Use case diagrams. Consequently, a relation expressing how the method rationale offered by the Actor method component contributed to the method rationale offered by the Use case diagram method component is added. For each of the identified method components we had to ask the team where in the development this artefact was used again and why it was necessary. This was needed in order to ascertain what the method component’s interfaces contained in terms of their prerequisite artefacts and their overall goals. In most cases the overall goal rational structure corresponded to the RUP. For instance, the Analysis class diagram fulfilled the goal of giving an overview of the early conceptual model for aspects in the system with responsibilities and behaviour. This method component was considered a prerequisite for the method
component *Design class*. A Design class provides a definition of a set of objects that share the same responsibilities, relationships, operations, attributes, and semantics and is typically used in the RUP for deciding what to implement as functions or information that needs to be stored in a database. Eliciting Design classes in the RUP involves analysis of the current Analysis class diagram.

Another aspect that became evident was that they omitted certain steps compared with the RUP. As an example, the RUP suggests that a method component called *Use case realisation* is performed in order to facilitate discovery of potential Design classes. The RUP also suggest that Design classes are collected in a *Design Model* component which did not have a counterpart in the PSU development method. These omitted method components usually function as a way to collect deliverables from a number of artefacts in order to create an overview and minimise the risk of forgetting results along the way. The method rationale they provide are also commonly considered unnecessary in projects of smaller size as these function as collectors of the sum of method rationale provided by other method components. Omitting these method components would not necessarily lead to an incorrect result, even though the risks for incorrect results would rise. We considered PSU as being a large project and thus possibly being in a situation where some aspects could, but not necessarily would, be forgotten. In the end, it all comes down to individual developers’ skills.

![Figure 33. Diagram describing the Method Components and their goal relations in the PSU project](image-url)
This reconstruction process was followed through until the method rationale of each method component was understood and their relations to other method components in terms of method goal rationale were identified. The PSU team was able to give account for their ideas and we could easily understand how they could reach rationality resonance with the help of their self-conceived systems development method.

The method components and their relations were documented in a diagram shown in Figure 33. It is an overview of the method components used in the PSU project. Each method component is named after the its deliverable artefact and shows how the artefacts contribute to realise the goals of the other method components. An arrow from a method component to another means that the former method component is considered to offer its deliverable artefact as a prerequisite to the latter.

7.2.1.3 Step 3
In Step 3 we compared the results so far with the rationality provided by the RUP. Comparisons between the reconstruction results and existing systems development methods are intuitively done also during Step 2 as this is almost impossible to avoid. Often we experienced that our understanding of the role of a certain method component was founded by relating it to how it is used in the RUP. As already mentioned, the roles the method components played usually matched each other. Studying the diagram one can compare the development process to the RUP. Even though the PSU project team already had adopted much of the RUP’s activities on their own, and had a development method that looked very similar to the RUP there are certain informative differences.

An example of how the processes differ can be found in the part that governs the Test discipline. In the RUP artefacts named Test evaluation summary, Test ideas list, Test result, and Change request are suggested. These artefacts can not be found in the PSU artefact diagram. Instead, they have replaced them with self-conceived artefacts covering much of the method rationale offered by the RUP artefacts.

The Test case template produced in the PSU development process is offering method rationale in the form of an ability to specify reusable test scenarios. This artefact replaces the Test ideas list from RUP which is intended to identify potential tests. In the PSU project they deemed the Test case template method component to be of greater value as input to their Test cases and found rationality resonance in that decision. They had a project that they felt that they could understand well enough to imagine that it would be possible to plan their test activities from a conceived set of templates rather than from individual lists as suggested by the RUP. If the PSU project decided to become more RUP adhering in the future, they could easily exchange their Test case templates for the RUP’s Test lists.
Similarly, the *List of measures* was considered to be sufficient in order for the team to record the results from the tests. This artefact also kept track of possible Change requests called for as a result of the performed tests and described the changes that would be required as well as ones that already had been done. Consequently, The List of measures replaced the Change request method component from the RUP.

The *Test result* was treated differently by the PSU project as they produced a Test log to keep track of the results from the tests and found it to be sufficient in terms of the method rationale it offered for producing a List of measures. The *Test evaluation summary* suggested in the RUP did not have a PSU counterpart at all.

Figure 34 describes the differences between the RUP and PSU’s reconstructed systems development method. Dashed boxes are RUP method components that had been replaced in the PSU project. The dashed arrows point to the position where the PSU team had replaced a RUP method component with a self-conceived method component.

The final result from Step 3 showed that PSU had a systems development method that corresponded to RUP in many aspects. In fact, the differences where so miniscule it would be possible to say that the PSU project already had adopted RUP with some adaptations.

As already stated The PSU project had not implemented ATUP yet but claimed that they planned to do so in the near future. As the consultants already had decided that PSU would not be chosen as a pilot project they did not worry about this fact at that point. Another aspect that gave the PSU project team some leverage was the fact that they produced results efficiently and were able to keep
the team working without ATUP. Hence, an implementation of ATUP would first be considered when it was time for the PSU project to fully adopt RUP.

Analytical work of this kind can be done to compare a reconstructed informal and/or implicit systems development method to the public method rationale offered by an externalised systems development method. During the course of a project assessment task a reconstruction of the existing development process would facilitate creating a clear picture of what really goes on during the complex business of systems development. Very often, the consultants explained, the tacit knowledge of the development processes was forgotten and the implementation was conducted with a starting point in previous experience from other implementation projects. Knowledge of this type would give the persons responsible for the method implementation process greater depth in assessing projects and a greater understanding of how the project develop systems today and how this process differ from the systems development method that is considered for implementation.

In relation to EME research, reconstruction activities containing Step 1 and 2 can benefit the understanding of how systems development methods communicate their method rationale in systems development practices. As our approach for method reconstruction from a method component perspective aims to clarify the goal achievement structure in a development project and make it more explicit, it will be possible to communicate an otherwise implicit systems development method to others. It will also be possible, by adding Step 3, to compare the reconstructed systems development method to the public rationality offered by another systems development method.

7.3 Conceptual Re-Modelling of the Method Component Concept

In this section we return to the method component concept we re-designed in chapter 6. The PSU case gave us an opportunity to put the method components in to action and apply them for a practical method engineering task. In the light of the results from the reconstruction of PSU’s implicit and informal systems development method it is possible to evaluate the method component concept and possibly make adjustments according to the design science approach described in chapter 2.

The method component concept was conceived through design goals introduced in chapter 4 and subsequently realised in a re-design in chapter 6. The design goals are a collection of properties that the method component concept is intended to possess. The following section is devoted to comparison between the original design goals and the experiences form the PSU project assessment. The comparison will be conducted by applying a conceptual modelling perspective much like what took place when the method component concept was formalised and re-designed in chapter 6.
Design goal 1. Self-contained: It must be possible to treat the method component as a self-contained part of a systems development method with regard to the guidelines that describe the deliverable and the process of producing such a deliverable. This way a method component can be modelled out from a systems development method or act as a building block when constructing new methods or during method integration.

Step 1 in our exploratory approach for method reconstruction consisted of an application of every artefact involved in PSU’s development process. This meant that by listing artefacts we would immediately identify potential method components as each artefact must be considered to be a deliverable offering an overall method component goal needed somewhere else in the systems development process. This also meant that the artefact perspective instantaneously ensured that our elicited method components would correspond to the chosen diagram level of granularity. Hence, we would not have to worry that the reconstruction result would yield method components that would be too small or too large. The reconstruction of the PSU development process showed that it was no problems to treat it as constituted of self-contained method components. The artefact level of granularity corresponded very well to how the team in the PSU project perceived their development process. At this time, we did not give any attention to exactly how each and every artefact was elicited. Hence, we did not make an analysis of the content of each method component. However, such an analysis could be performed with a starting point in the method component diagram in Figure 33. The reconstruction also showed that the method components conceived by the team itself easily can be replaced by components from the RUP if needed which indicates that the idea of a method component as a building block is viable.

Design goal 2. Internal consistence and coherence: The method component must be perceivable as an internally consistent and coherent entity. Another way to put it is to say that the construct should be stable over time and without lose ends. As a result the method component will be perceivable as meaningful. The principle is an ambition to create congruence amongst method components and homogeneity in systems development methods.

Treating every artefact in the PSU development process in the same way during the reconstruction let us adhere to the congruence aspect of the method components. They are essentially similar to each other in terms of properties but the contents and method rationale differ and give them individual characteristics. However, since we did not conduct analysis of the content of each method component it is difficult to evaluate this design goal completely.
Our reconstruction process was more focused on the external view of method components which directed us toward the method components’ interfaces rather than their full content which is found in their internal view. Nevertheless, the goals that can be found in the interfaces along with the deliverable artefacts are considered internal parts of the method components. These aspects were focused in every method component. Additional analyses of other internal aspects are possible by focusing on the artefact since the deliverable artefact is a container for the results achieved while using the method component and ‘using’ means that a number of concepts are put into action and a defined notation is used to keep record of the results. It can be considered to be a process of reverse-engineering. Since the deliverable artefact contains much of the method component content indirectly, it is possible to apply our approach for method reconstruction and still claim that a method component perspective is upheld.

**Design goal 3. Rationality:** Actions are performed and prescribed for reasons. Thus a method component must have an identified target state, a purpose for its existence. The arguments are motivated by the components inherent method rationale and address the goals and values that the method component realises.

Step 2 during our method reconstruction is where this design goal was primarily satisfied. During this step we analysed each artefact in our artefact list and re-defined them as method components. The transition between a strict artefact focus towards method components is easy since it more or less only involves the definition that there must be a method component with a content that is resulting in each artefact. If there was no content, there would simply be no artefact either. The artefact also carries the offered overall method rationale of the method component in line with our design goal of rationality since it expresses a purpose for its existence along with its goals in the interface.

Applying a method component perspective let us create a view of PSU’s development process with method rationale in the foreground as Step 2 required an analysis of the goal rational structure that lies behind and motivates PSU’s development process. Each method component was recorded according to how it contributed to the realisation of goals in other method components through relations to other method components expressing method rationale. The diagram describing the identified method components and their goal relations in Figure 33 shows how we structured the initial list of artefacts into method components.

The reconstruction of PSU’s implicit development method also showed that it is possible to classify and record self-conceived method components once their overall goals are identified and understood in relation to their deliverable artefacts. The PSU team members had a development method with a method rationale that was very similar to the offered by the RUP, but they also hade made some adjustments and replaced some of the components in the Test
discipline with self-conceived method components. These latter method components need to be analysed and their method rationale must be understood during Step 2. Otherwise, the reconstruction process might yield a picture that does not correspond with what is actually taking place during development activities. The reconstruction process showed that it was possible to focus and record the method rationale of self-conceived as well as standard method components.

Design goal 4. Connectivity: The method components must be possible to connect with each other. Each method component, viewed as part of the systems development method, should contribute to a chain of goal achievements adding to the overall goal of the specific project. Thus method components must have the capability to express how they are connected to each other and how these connections can contribute to the overall goals linked to a specific project.

The focusing of shared rationalities through method components interfaces during Step 2 directly led to issues concerning the connectivity design goal. The process of finding relations between method components by the rationalities they offer as deliverables clearly show us that it is possible to connect them in this way and thus fulfilling the design goal. Step 2 in our approach and the transfer of perspective from artefacts to method component interfaces was perceived, by all involved persons (project team, consultants, and researchers), as a natural way of treating the collection of artefacts once they were re-defined as individual method components. This showed that the design goal of connectivity proved to have been successfully implemented in the method component concept design.

Design goal 5. Applicability: It should be possible to map the method component construct onto any existing systems development method in any situation.

During the reconstruction process we successfully applied the method component concept on an implicit systems development method. We also treated modules from the RUP as method components during Step 3 where we compared the method rationale of the PSU’s Test components to the Test discipline in the RUP. This involved method component application on part of the RUP as well. Furthermore, we were able to maintain a method component perspective when analysing method components that were self-conceived. These results suggest that we have no new reason to doubt the method components’ possibilities for application.
The conceptual analysis above does not imply that we did not make any alterations on the method component concept at all. We actually realised one aspect we had not previously considered. As already stated above we did not make a complete analysis of each method components’ content. However, when we discussed the analysis results with the PSU team members we ended up in a discussion about Actor roles for some of the components. In some cases the Actor roles could be considered to be either carrying out or participating in a certain action. There would always be an instantiation of a <Carry out> association between Actor roles and Action since an Action demands an actor. This is not the case with the <Participate in> association. This is already defined in the UML diagram through the choice of multiplicity for the two associations. What needs to be added is a definition that states that an instance of an Action class, i.e. an Action object can not have two links to the same Actor role object. For example, the Actor role class can be instantiated into two Actor role objects, perhaps a System analyst and an End user. An Action class can be instantiated into a Draw activity diagram. In that case it is necessary to specify that the Draw activity diagram object can only have one of either links to each Actor role object. If an instantiated Action object has two or more links to Actor role objects, the links can not be allowed to lead to the same instantiated Actor role object. Hence, a System analyst object (of the Actor role class) that has a <Carry out> link to Draw activity diagram (of the Action class), can not have a <Participate in> link at the same time. This constraint is defined by using the Xor-constraint which is subsequently added to our UML diagram.

![Figure 35. The Method Component Concept v1.1](image-url)
7.4 Method Rationale - Lessons Learned

Method rationale defined as the goal and value structure that motivates a systems development method is something that is already incorporated in the method component concept through its Rationality design goal. Nothing in this case study indicated that we needed to make any alterations in the definition of method rationale as it is defined in chapter 4. However, the concept of method rationale plays additional roles during a method reconstruction task and the learning experiences from using method rationale in this context are worth noting.

In the PSU case we often used the concept of rationality resonance to identify and understand how the PSU team members found method support in their self-conceived method. An example of this could be found in the discussion about differences between the reconstructed PSU method and the RUP. In order to understand how the Test case template and the List of measures gave support within the PSU process we first of all had to understand how the PSU team members claimed they got support from these method components. This means that we had to match their private rationality to what was actually being done in those method components. This means that we as researchers in EME use the concept of rationality resonance in order to interpret the reasoning that motivates implicit systems development methods. The ability to focus on rationality resonance during EME research seems to have great benefits as it gives the researchers a possibility to judge aspects of systems development maturity and awareness.

The reconstruction of PSU’s implicit systems development method showed that method rationale as it is defined in chapter 4 is possible to incorporate in the method component concept. We did not experience any faults in the way that method rationale has been connected to individual method elements in the method components’ content part. However, it became even more evident that the problems related to modelling the value anchoring structure of a systems development method introduced in section 6.5.2 might be correctly observed. An analysis of the value anchoring structure is difficult as the structure is dependent on the user. In a normal systems development practise it might not be a very common situation where a method component’s underlying values might lead to problems. There was such a method component; the goals this underlying value aimed to achieve would probably appear questionable. Who ever heard of an immoral method component? Nevertheless, we once again must underline the fact that every method component has underlying values that make the goals worthwhile aiming for. However, making analyses of the underlying value anchoring structure during much of EME research seems to be of little practical use though it is possible to imagine situations where such analyses could be beneficial. One example is a situation where a new systems development method is constructed with an actual intention of being value based. One might think of the programmers’-liberation-from-tedious-
7.5 Concluding discussion

The PSU case study gave us an opportunity to explore the potential of the method component concept in a practice setting. EME research has to cover research in all three method application areas as they are described in Figure 14 in chapter 4. This case would probably fit best in the method reflection sphere since it covers activities where we as researchers are helping method users to reflect on their own development processes. Going back to Figure 14 we can also see that the reconstructed systems development method of PSU is possible to use as feedback to the method engineering sphere. The process of reflecting over the constituents and structure of the implicit method used in the PSU project also gave the involved team members an opportunity to reflect on their actions when they used the method in action in the method-in-action sphere and to communicate these reflections to each other and to others.

Using the method component in method reconstruction of PSU’s implicit systems development method proved possible. In addition to just applying a method component perspective on the implicit systems development method we also explored the possibilities of formulation of steps in order to conduct a method reconstruction task. This very simple approach consists of two basic steps where the implicit method is first analysed by the artefacts produced along the way. This artefact focus gives a correct level of granularity for the method components when each artefact is subsequently re-defined into a method component. The second step gives attention to the method components’ interfaces and defines how the method components are related to each other by the goal rationale offered by their deliverable artefacts. The suggested following steps only concern situations where the reconstructed systems development method needs to be compared with another explicit method or if certain method components need to be added or removed.

The applied approach for method reconstruction was used along with the definition of method rationale. This gave us a possibility to compare rationalities between method users, the reconstructed PSU method, and the RUP. Our experiences during the method reconstruction case study made us realise that the practical use of analysing the value anchoring structure of a method’s rationale might be of lesser practical use, even though it is possible to identify situations where analyses of this type might be useful or even necessary.

Another lesson learned during the time we did the method reconstruction task was a need to conceptually re-model the method component concept. The re-modelling effort consisted of an addition of a Xor-association in order to define
an identified constraint on the double associations between Actor role and Action.

Our results indicate that the assessment phase can benefit from applying a method component perspective during tasks that involves reconstruction of the existing development process. If the development process is implicit, as was the case with the PSU project, the method component concept can help to create a reconstruction of the involved artefacts and communicate their roles in the project. The suggested approach allows comparisons with the systems development method considered for implementation and helps method implementers to an enhanced understanding of, and a possibility to communicate about, the development situation.
8. Method Rationale in play – a Swedish Armed Forces Case Study

The concept of method rationale is something that should not only exist on a theoretical level. It is very easy to think that this is yet another concept that researchers have invented to point the finger onto something “new” but still utterly useless when it comes to practical usage. The ambition with the concept of method rationale and the possible benefits of its explication must be thoroughly studied in practical and very real situations. Only when this is done, will we be able to ascertain if the proposed concept can live up to its words and be used in EME research. The field study in this chapter takes this problem into consideration by applying the concept of method rationale to a real life situation. It consists of a case study analysis where I study how the concept of rationale is communicatively present during actual method usage in a group of method users. This is done by exploring how method rationale is used to direct attention, create consensus and review results in a modelling team. The research presented in this chapter can be regarded as belonging in the method-in-action sphere in Figure 14. It is a very obvious example of how traditional method engineering can be extended into areas of research it previously has not been applied in, namely for studying social method-in-action. The extension is considered to reside in the fact that the conceptual foundation for the case study is founded in method engineering concepts that are applied as interpretative frameworks.

The case study took place in a method project within the Swedish Armed Forces and concerns observations during method usage situations. The main goal during the studied modelling sessions was to explore a hypothesis that the RUP (Kruchten, 1999) would be possible to use for describing the military business in a way that would not only capture relevant military specific codes and regulations, but also provide a ground for future systems development.

The benefits of this study can be identified through exploration of the role of method rationale in the discussions amongst the group members involved in the modelling sessions. If the concept can be found present and its role be understood it will validate the claims that method rationale is an integral part of method communication and that it is something that needs to be considered when defining what a method is constituted of. The research approach during the case study will be interpretative as I am trying to understand the role the dimension of method rationale plays in a modelling scenario.

8.1 Chapter Overview

This chapter is a description of how method rationale is used in communication by professionals in a real systems development method application situation. It starts with a description of the case background focusing on problems identified within the Swedish Armed Forces and how they try to solve these issues. The case study background is followed by an introduction to the task that was
presented to the members of the modelling seminar prior to the event. A description of the qualitative research approach comes next. It lays the foundation for the analysis of the seminar and describes the position I have taken regarding the choice of case study type. I also describe how I view method rationale as a theory in this specific case study context. Then follows a discussion where I describe the generation of my core categories and how I have collected and analysed my empirical data. The section that follows the description of the chosen research approach is a presentation of the analysis. The analysis is presented in a way that illustrates the three core categories I have found. The chapter ends with a conclusion and lessons learned concerning the role of method rationale. This leads to an increased understanding of the method rationale definition.

The entire chapter is captured in the activity Empirical Grounding Swedish Armed Forces that can be found in Figure 4. Overall description of the research process. Input as well as output related to this activity is the method rationale definition. This means that my interpretations from the case study expand my understanding of the concept of method rationale. It also means that the definition itself might be reformulated if my results indicate that it would be necessary. A figure describing the chapter details can be found below.

Figure 36 Chapter 8 details

8.2 Case Setting

In recent years the Swedish armed forces have been subjected to budget cuts due to the end of the cold war. In 1976 the budget was 3.1% of the GDP. In 2005 the Swedish armed forces have a budget equivalent to 1.52% of Sweden’s GDP. Since then the Swedish, mostly conscription, army has turned to peace keeping operations in cooperation with the UN, European Union or NATO. Swedish Battle groups are currently deployed in 15 locations throughout the world including Afghanistan, Bosnia – Herzegovina, Lebanon and Liberia. (Figures are taken from the official web page of the Swedish Armed Forces http://www2.mil.se/sv/).

In accordance with new demands and a diminishing budget, The Swedish armed forces today have a much smaller organisation than before. In an attempt to maximise the resources and improve overall efficiency there was an attempt to create what was referred to as a Network Based Defence. The armed forces imagined a method to “…create capability to fast, flexible and coordinated efforts, both in Sweden and abroad.” The idea was to create a modern military organisation which was highly coordinated internally and with other government agencies such as the police and the fire department. The solution should involve modern ICT (Information and Communication Technology) based systems and
for that reason there had to be a consensus in the choice of method. A lot of the government contractors delivering ICT solutions to the Swedish armed forces are using specific systems development methods, typically RUP (Kruchten, 1999). Therefore, the Armed Forces thought it would be a good idea if the systems development methods could actually be used to capture the unique circumstances in the military. If so, then it would be recommendable if as much as possible of the Swedish armed forces’ operation would be described with RUP as well. That way, it would be a fairly straightforward task to start actual systems development projects when needed since a lot of the initial analysis of the organisations business would have already been covered. One could also count on that the business descriptions were somewhat stable over time as it is a military organisation where virtually everything is conducted through adherence to orders and codes.

This idea ultimately led to the start of a project where methods for military and systems development projects where analysed in an attempt to find corresponding benefits. As the military industry favoured RUP as a way to develop systems the project found that it RUP would be a natural starting point for finding methods capable of capturing the operations of the Armed Forces. That way the Armed Forces would not have to develop new methods. Instead, they could identify discrepancies between the original method and their method needs and adapt or complement what seemed to be missing. A five day modelling seminar was scheduled where this hypothesis would be tested for the first time.

8.2.1 The Modelling Task
The purpose of the modelling seminar was to ascertain if RUP/UML was useful for describing business aspects of the Armed forces. The modelled scenario
concerned establishment of information infrastructure and information assets during a military deployment in a peace keeping operation. The actual modelled scenario will not be enclosed or described in detail as it is considered to be classified information.

The case study was conducted during a modelling seminar over five days in a Swedish military facility. A total of eleven persons were involved, not counting the researcher. The group was a mix of systems development professionals from both the civil sphere and as the military industry. Liaisons officers were also present as providers of information concerning the analysed business.

The task for the modelling seminar was to test if the Business modelling discipline from RUP would be possible to use to describe the business carried out by the Liaisons officers in an initial set up phase during a military deployment. Basically, the test was founded in a hypothesis that it would be possible to use RUP for the above stated purpose and the hypothesis would be falsified if it turned out that RUP couldn’t be used as planned. The person responsible for leading the seminar (henceforth referred to as P) has had a key role in the development of RUP and he has worked with RUP as a user and method consultant for many years and has to be considered an expert.

8.3 Interpretative Research Approach
The epistemological standpoint when it comes to case studies can vary. Case studies can fundamentally be of either positivist (Yin, 1994; Benbasat et al., 1987) or interpretative nature (Walsham, 1995). This case study is of the latter kind since I have an ambition to study and analyse the communication between people and try to make sense of their discussions. Data collected during the case study are regarded the researchers own subjective interpretations or constructions of other people’s interpretations or constructions (Geertz, 2000). This means, that I as a researcher must be aware of the fact that the observations cannot be done objectively. They will always be subjective interpretations of what is going on. To handle this problem, a researcher’s ambition is to give a description “thick” enough to ensure that the following analysis will have a more solid foundation (Walsham, 1995; Geertz, 2000). This means that I will have to describe situations that arise during the modelling seminar in an intelligible fashion to show transparency for my conclusions.

As the object of study will be communication of method rationale, this concept will be in the centre of attention in order to capture the relevant social dynamics in the scenario (Dyer Jr and Wilkins, 1991). This means that I will focus on and describe statements and actions that can be considered being about method rationale. This leads to the question of my role during the modelling seminar. Traditionally, a researcher can take an outside role or choose to be more involved (Walsham, 1995). The degree of involvement can be high as in action research (Baskerville and Wood-Harper, 1996) where the researcher actively participates in finding the solutions. Since I did not want to influence
the discussions I chose to take an outside role during the modelling scenario through observation. This means that I was present at all times but not contributing to the solutions. Instead I was taking notes, observing and trying to interpret what was going on.

As described in chapter 2, according to Eisenhardt (1989) there are three distinct uses of theory in case studies. Firstly, theory can serve as an initial guide to design and data collection. Secondly, theory can also serve as part of an iterative process of data collection and analysis, and thirdly it can be the resulting final product of the research. This case study will analyse how method rationale is present or unconsciously inherent during method labour. Method rationale will be treated as a theory or a theoretical starting point for my analysis. Hence, the use of theory will follow the second use of theory according to Eisenhardt. To yield the desired result as part of the iterative process of refining the concept of method rationale it is important that I am willing to modify the initial definition of method rationale if needed (Walsham, 1995). The person responsible for leading the modelling seminar had not been given any instructions or been introduced to the concept of method rationale or method components. The case study aimed to explore how method rationale was naturally present during method communication. Obviously, any prior information of the method rationale concept had to be excluded.

8.3.1 Data Collection and Analysis

During the modelling seminar I was sitting along as an observer. This means that I did not contribute to any solutions nor did I give any instructions, directions or requests concerning how they were supposed to think, communicate or interact. Everybody knew I was an outside researcher but the only thing they knew about my task was that I was present to study how they used the method and to comment on how their approach to method adaptation compared to established research. I gave some brief comments on their chosen adaptation approach in relation to research during the seminar as they asked me. They later on received a report describing my perceptions during the seminar. This report focused on their chosen adaptation approach and how it related to the two typical approaches to method adaptation; the contingency approach (Fitzgerald et al., 2003) and method engineering (Brinkkemper, 1996).

The data was collected through written notes. Apart from notes that were very general, i.e. descriptions of the current task, I also made notes of issues that could be related to method adaptation. However, most of the notes were about communication and presence of method rationale. Notes were taken of statements and discussions involving realisation of goals, how to think, purpose of activities, needs, method semantics, analytical agreements and disagreements as quotations and descriptions of the situation and the behaviour of the participants. It basically means that I reacted to and focused on situations where there was a presence of the knowledge dimension of methods, either inherent in
the method or in statements or actions. The data was then analysed and coded (Strauss and Corbin, 1998). The process of coding was divided into three phases. I started with an analysis of the written material I had collected on a statement level. This means that I was not focusing on my empirical material concept by concept. I rather tried to interpret the meaning behind various statements or situations. After this initial coding I had a compilation of descriptions of statements and situations that I felt related to method rationale in various aspects. The second coding phase tried to make sense of the role method rationale played during the modelling seminar. I therefore viewed my compilation with a different perspective and tried to find meaningful categorisations of the various roles that method rationale played. After just a short period of working with the categorisation phase I felt that three typical communication characteristics had arisen where method rationale played a vital role. The three identified communication characteristics were:

- Method rationale as attention director
- Method rationale as consensus builder
- Method rationale as check point

The third phase was a verification phase where I made sure that every statements and situations I had compiled could be said to belong to one of the identified communication characteristics. I also returned to my original empirical material to check if these three identified communication characteristics could be used to find statements or situations I had not initially found to concern method rationale. At this time I experienced saturation since nothing new showed up. I felt that the empirical material no longer could give me any insight apart from what it already had. Consequently, these communication characteristics became the core categories I am using to describe how method rationale is used in real situations. The core categories and the arguments for them are presented in the following section.

8.4 Rationality in Play

“The biggest risk when adopting RUP is that you do not know what you are doing”

This quote, uttered by P, describes the ambition to let the seminar have a clear presence of “knowing why” and an “understanding” of what we were supposed to do. An analysis of this statement in relation to method rationale would say that there was an ambition to let the rationale of each method activity undertaken to be known. Thus, this ambition indicates that method rationale is
something that is fruitful to consider when working with actual development projects. Not considering this knowledge dimension of systems development methods is actually described as risky.

This statement, of course, means nothing if method rationale is not clearly present during. It would only prove to be another cliché about something that doesn’t hold any real value. This section will be devoted to an analysis of the seminar, focusing on the role method rationale played during the five days of intense modelling.

8.4.1 Method Rationale as Attention Director

The seminar started with a description of what they were supposed to achieve. The task involved a Business analysis using RUP as a test to see if it could be used to describe the unique conditions of the Armed Forces. As the group consisted of several people with various backgrounds P commenced with a description of the overall goals with Business modelling. This description focused on the benefit of using Business modelling for this specific purpose. P described what was gained by analysing the aspects covered in Business modelling and how this was related to developing systems in general, when developing a specific system addressing the issues found through analysis during the seminar and the task of evaluating RUP as a feasible method to capture the unique conditions of the Armed Forces. This was a typical example of how the inherent method rationale of a systems development method is used to guide the attention of a group put together to solve a particular problem. Sometimes the focus can tend to deviate from the original task and the idea to regularly return to the overall method rationale seemed to be a good one since not everyone had knowledge of RUP before. Some of the participants came from FMV (Swedish Defence Material Administration) and they had already adopted RUP for systems development projects. Basically they were present to see if the produced results would be something that they could use for developing systems. At times it became obvious that these people were focused on technical problems and realisation issues as they raised a lot of questions about certain technical details. It was clear that they were thinking about how they would develop a certain system or capture certain information in an already existing system. To direct the attention of the group, P was keen on returning to our original agenda. This was a test. We were supposed to focus on the task at hand and not wander off mentally to actual systems. One way of doing this was to start every session with a review of the purposes of the current method component.

A common critique towards methods in general is that they tend to result in overproduction of documents (Parnas and Clements, 1986; Introna and Whitley, 1997; Beck, 2000). During the seminar, situations arose where especially military officers questioned why the group were producing all these documents.
Officer: “Why are we producing all these documents? Does anyone really need them?”

*P* replies: ”A good document is a good complement to face to face communication”

He then continued to describe how documents played a vital role in an iterative work pattern. By documenting the previous results one can have the possibility to review work already done and improve upon things that seem to be faulty or inefficient. By not recording the progress in some way, there is also a definite risk that time is devoted to unnecessary labour as things that already have been analysed and covered has to be done perhaps multiple times. By having this discussion P emphasises the need for documentation during method use. He emphasises why we produce the documents and explains some of the underlying method rationale for this. The document was to be considered an aid to communication. It did not only serve as a record of solved problems. It rather was a way to facilitate communication about what had been done, what needed to be done and what should have been done differently. By explicating why they produced the documents P helped the group to understand what they gained by doing this. There is a risk that documentation is given too much emphasis and that the actual problems that need to be analysed are forgotten. P directed the attention toward the task rather than toward the documents and reminds the group of what they were supposed to do.

Another example of how P directed attention to the method rationale of the method component being used was to exemplify the questions that were supposed to be asked during modelling in order to ascertain that the method component being used could deliver the desired results.

*P*: “Is it possible to reach the desired goals we have with this diagram?”, “Will this activity help us to reach our goals?” or “Can Business modelling express the service dimension in a way that is desired?”

These types of questions clearly address the overall method rationale. P reminded the group to think about whether they could achieve the goals they wanted to achieve by the method rationale delivered by the method components in the Business modelling discipline of RUP. They were also directed towards the question whether the method components could help them achieve the goals they wanted to achieve. One might think that these two focuses are the same. However, they are not. The first one takes a starting point in business needs and analyses whether the desired method rationale can be found in the proposed method component. The other focus takes a starting point in the method
rationale delivered by the method component and analyses whether it is something that is useful to fulfil the needs you have in the present situation.

The method components used were not only analysed whether their overall method rationale was deemed useful. They were also analysed internally when needed. At times P asked:

P: “Does this diagram need added semantics?”

The question addresses the internal method rationale of a method component, sometimes hidden somewhere in the network of goals that is inherent in each method component. This is done by a direction of focus on the modelling primitives of the method component being used. The progress is recorded in documents and models describing how the members of the modelling seminar perceive the business’ reality. The documents and models are made up of different modelling primitives who together contribute to the overall method rationale of the method component. Going back to the definition of the method component in chapter 6, the individual modelling primitives are method elements and carry their own method rationale. For instance, the concept of ‘actor’ is something that is deemed rational to know of and to record in certain method usage situations. This means that a graphical representation of the concept in the form of a symbol is a reference to the original concept and the rationale behind it, i.e. drawing a symbol in the form of a small man when modelling a Use Case diagram I RUP is not something that is done because it is fun. It is done because the concept of the ‘actor’ and the rationale behind identifying instances of this type fulfils goals that can contribute to something else. The actors, for instance is a way to find users and user groups of an information system. It is an example of how P directs attention towards method rationale at a very low level of granularity as he asks about the method rationale of very specific method elements. He basically invites the group members to think about the fundamental design decisions connected to the method component and to ask themselves if the method rationale these decisions provide is enough, or need to be enhanced through addition of new symbols which refer to and can be used to capture necessary concepts.

This invitation is also complemented with a request from P that the participants of the modelling seminar try to think outside the box and instead regard potential limitations as possibilities.

Thinking outside the box is not an easy task, especially if you are supposed to do so, on demand. To help the members of the modelling seminar to think outside the box, P gave some insight to how one can think;

“You are looking for the typical bullseye name for things! Is this what we are looking for?”
He continues to describe the need to have an ability to express different levels, criteria, properties, etc. in a way that makes sense in the business context. Is that possible in our case or do we need to add new concepts and corresponding symbols?

8.4.2 Method Rationale as Consensus Builder

The previous section described how method rationale was used in communication as a way to direct attention. This was primarily a unidirectional form of communication where one particular individual was trying to teach or coach someone else in how to understand the underlying method rationale of a method component. In these teaching situations it is customary to ask, like P did, if everyone has understood the meaning. Even though the members of the modelling seminar replied to these question and thus communicated back the main knowledge transfer of method rationale was unidirectional. P explained the method rationale behind what we where doing and received information whether he succeeded in this task or not.

A true multidirectional form of communication of method rationale can be found in the situations when decisions were taken concerning how a certain business phenomenon should be defined or understood. They are examples of how method rationale is used to build consensus in a group during a modelling task. Basically, how things are agreed upon.

At one time the group needed to make a decision which of the activities they needed the most help with in the scenario being modelled. This discussion started in the perceived needs of the present military officers. A discussion arose where they started to discuss which parts of the scenario that was the most complicated and thus needed to be addressed first. Some of the officers wanted to model a situation where the responsibilities for establishing liaisons function needed to be modelled. Others, mostly from FMV found it necessary to define standard protocols for the various types of communication that could occur. The two groups could be described as expressing two different private group rationalities. Of course the rationalities were not private in a true sense since they were shared between different individuals. However, in relation to their possibilities to achieve rationality resonance, it is necessary to think of a group sharing the same goals as having a shared private rationality, a kind of collective method rationale. The desired goal for the officers in this situation concerned their wishes to reach a target state where responsibilities in relation to certain business scenarios (Business use cases). The proponents for analysing the protocol standard aspect wanted to reach for a goal which included technical definitions which would make it easier to understand the required level of performance from communication systems and their interfaces. These two group rationalities were compared through an open discussion and moulded into a joint decision. In the end the group decided to model the responsibilities for establishment of liaison functions. This decision formed the desired collective
rationale of the group. The collective rationale was then matched to the method rationale offered by the method component. In this case the Business actor method component was chosen since its overall goal aimed to clarify the responsibilities different roles had in relation someone else or something in the business environment. Hence, the Business actor method component’s public rationality had a possibility to create an arena where the private collective rationality of the modelling team could achieve rationality resonance. This was an example of how a group could create consensus through communication of method rationale.

Group discussions rarely pass without any conflicting opinions and this modelling seminar was not an exception. At times participants from FMV wanted to discuss very technical issues that were outside the scope of the task. Since they modelled a business’ need for a tentative system they immediately started to think about how this system must look like to be able to work with already existing systems. Those types of questions were not supposed to be answered during the modelling seminar. The seminar aimed to explore the possibilities to use RUP to describe the Armed forces business in a fashion that would make sense to FMV. FMVs role was to deliver input to the question whether the modelling result would be something they could use for systems development and not to questions concerning how the system modelled must be designed in order for it to be able to communicate with already existing or planned systems. There was a clear difficulty in keeping these two perspectives apart, not only for the participants that came from FMV. These situations were generally solved through communication where the group returned to either the rationale provided by the method component or the rationale they had agreed upon earlier. Group decisions are, in the best cases, the result of consensus. Sometimes part of the group does not agree with the decision and matters have to be resolved by some sort of majority ruling. During the studied modelling seminar I never experienced anything similar to a vote or enforced decisions. Most discussions were very open, candid and straightforward. At one time the participants of the seminar were discussing how certain activities “really” were performed. The activities in question were regarding who actually was responsible for producing the service called “Are my liaisons requirements covered?” In order to ensure progress they had to decide which kind of tentative design they would need of a system to ensure that it would be capable of delivering the necessary information. An officer explained:

\[O1: \text{“We will most likely have a situation where liaisons troops acting like a spider in the web. They will deliver and serve all communications infrastructure.”}\]

Two other officers replied:
Others fell in and the group could reach an agreement upon the system they would model. Given this consensus, it became necessary to specify some of the capabilities of the system, at least when it came to issues such as interfaces and technical restraints. This time, however, the participants from FMV did not become over-technical and they did not want to discuss issues outside the scope of the scenario. Instead, they conformed to the rationale of the others and were able to give suggestions and input of the desired type.

During these discussions about what the group wanted, P can not be said to have influenced the joint decisions in any major extent. His role was rather to help by giving directions when the rationale of the current method component was not clear or when the group needed directions in how to prioritise. It is important to note that P did not want to take part in prioritising. He rather wanted to help the group to understand the rationale offered by the various method components so that the group could understand what they wanted to do in order to realise the rationality manifested in the joint decisions in the group. As a result, the communication on a group level had a very strong foundation in both the method rationale offered by RUP and the rationale that was produced in joint decisions. Conversely, and even more importantly, the rationale produced in joint decisions and the matching of this rationale against the method rationale offered by the RUP had a solid basis in communication.

**8.4.3 Method Rationale as Check Point**

During any modelling task it becomes necessary to review the results so far. This modelling seminar was not an exception. In regular intervals, P stopped the task and asked the group what they thought about the progress so far. This also occurred after natural pauses like coffee or lunch breaks. The focus during these interventions lay on both the quality of the results and the primary goal of the modelling seminar which was to test whether the RUP was useful for describing the operations of the Armed forces.

My focus during the case study was to explore if method rationale was used in communication in real life situations. An important aspect of this focus was to study the actual results in conjunction with communication. Questions arose whether the group had produced the desired results and if they where useful. I found these types of questions to be particularly interesting when they were related to the concept of method rationale. The reason for this was that it gave the method user a possibility to think about whether or not he had accomplished what he set out to do. In order to do so he had to review the rationale of the goal of the task, what kind of phenomenon he was trying to capture. Secondly it also
gave him an opportunity to return to and review the method rationale provided by the method component he was using. By comparing these two perspectives on the task, the rationale behind the task and the method rationale, he can make a judgement. The judgement will consist of two important parts. Firstly, he will make judgement on whether the method rationale of the method component was able to help him in the situation where it was used, i.e. did it match the rationale behind the task? Could he do what he wanted to do? Secondly, and perhaps even more important, he will have new experience that will add to the already existing knowledge he has of method use in general and of the used method component in particular. Working in groups, as they did during the seminar also added another dimension as the group format gave an opportunity to discuss and compare individual judgements through communication with the others. In this case study, these review situations were quite vivid and “talky”. The participants compared experiences with each other and commented on the results. This gave them an opportunity to compare their views on the task and learn from each other. The group discussions proved to be a function where method rationale was very clearly present. In these discussions, some of the participants became aware that their rationale concerning the method component or the rationales behind why they were modelling the specific business activity was different from the rest of the group. It gave them a chance to realise and learn from their previous misconceptions. An example of this can be found in the situation where an officer suddenly realised that the meaning behind a certain concept. He realised this during a discussion where they were reviewing the progress so far. As a result of this realisation, the group became involved in a discussion concerning the need to translate military concepts and functions in a way that would make sense in a later systems development project. The group also realised that organisational and staff requirements had to be included to the adapted version of the RUP for the Armed forces. Hence, the group’s individual rationale evolved into consensus or a collective rationale.

Another example of how communication centred round results showed presence of method rationale is a review we had of a class diagram. The class diagram described information concerning communication at battle group level consisting of approximately 3000 men. At this time some of the officers did not realise how this somewhat abstract diagram would be beneficial. P then found it necessary to return to the method rationale of class diagrams and explain that this type of information was a model of the information that the Armed forces would need in various situations. P then continued to describe various situations where specific information was needed and how this class diagram could be used by FMV to emphasise the benefit of producing that particular document. During this time, some of the officers previously not accustomed to class diagrams reported that they experienced a moment of revelation as some of the things undertaken during the seminar suddenly did not feel abstract or unnecessary anymore. They were also able to realise that the produced results
would be beneficial for their purposes as they could see which kind of information they could extract from the planned system. The officers realised that the rationale behind their needs, i.e. “why do I need this specific information?” could be served by the method rationale provided by the method component “Class diagram” as it is a component with this particular ability. The class diagram is most useful for this purpose. This was clearly an example of a typical check point scenario as the concept of rationality was scrutinised from a method component view, as well as from a method user view. The check point resulted in not only validated results, but also in valuable lessons learned for the group members.

The situation described above was mostly about how relatively inexperienced method users use method rationale in check point characteristics. When it comes to more experienced method users who have previous experience in using a particular method these types of characteristics seldom occur. For these persons the check point characteristics have a different meaning. For them, the review is more about checking the quality of the results rather than to check whether they have understood the method rationale correctly. During the seminar the group had participants from FMV who had previous experience in RUP. For these people, the reviews had a different meaning as they already understood the underpinning principles of the RUP and the method components being used. Instead, they could judge the results based on their quality and usefulness. This judgment was connected to their own rationale, i.e. what they wanted to accomplish with the modelling. Since they already had previous knowledge of RUP and understood the method rationale behind the components being used they could pass judgement whether or not it had been a good idea to use that particular method component and if the group actually captured what they set out to capture. The multidirectional communication between the participants who had this knowledge proved to have a strong foundation in method rationale.

A participant from FMV thinks out loudly:

*FMV1:* “OK... This diagram is supposed to show us the information we want, or need rather... I can see how this system can work with what we are contemplating at FMV, although we need more specifications.”

*P:* “Yeah. We will complement the templates with your specifications at a later stage but I can make a note of it so we won’t forget it.”

*FMV2:* “I’m not sure if it was such a good idea to design a new database for this purpose... Wouldn’t it have been better if we could have used an already existing specification?”
O4: “Let’s not forget what we are trying to do here! We are not involved with actual business modelling or systems development. We are just supposed to test the hypothesis that RUP can be used for our purposes.”

FMV2: “Yeah right... Well in that case it all looks OK. It’s a good enough model of the information for the scenario”

This discussion clearly shows that the participants from FMV had prior knowledge of the method component being used and they compared their own rationality with the one offered by RUP. The discussion also shows how the participant FMV2 momentarily had lost the insight in the rationality of the seminar. These types of multidirectional communication characteristics where method rationale was used to review the produced results were common and very useful as the group could identify possible errors or misconceptions that had gone by unnoticed. Focusing on method rationale and comparing it with the rationale behind the task proved to be the way that the results were reviewed.

8.5 Method Rationale – Lessons Learned

The purpose of this chapter has been to explore different roles method rationale play in practical method usage situations. Applying a method rationale perspective on what takes place in the method-in-action sphere has enabled an enhanced understanding of the concept itself and how case studies in EME can be conducted within this sphere.

My interpretations during observation of the modelling seminar led to a categorisation of three distinct communicative roles that method rationale plays in the method-in-action sphere. The case study showed how the leader of the modelling seminar used method rationale as a unidirectional Attention director when he wanted to emphasise the private rationality provided by individual method components. Method rationale also served as a multidirectional Consensus builder when the group took joint decisions, and as a multidirectional Check point when the group reviewed their results.

In chapter 4 method rationale was defined in relation to just one method and just one method user. In chapter 6, the method component was re-designed and suggested as a formalised method module concept for EME research. This involved definitions of how method rationale is manifested in the individual method elements that constitute the content of the method components. Chapter 6 also covered how a complete systems development method can be modularised into method components and how these method components can be connected to each other in a chain of goal achievements. This case, conducted at the Swedish Armed Forces showed that it might be necessary to specify situations where the collected private method rationale of a group of method users might be necessary to collect into something that can be referred to as
**collective method rationale.** This has consequences as it imposes new requirements for how the rationality resonance phenomenon can be understood in situations where EME research is conducted on groups of method users. This directs the attention to the model describing rationality resonance in Figure 17 as it needs to be evolved to reflect this aspect.

Thus, we add a class for describing groups as constituted of 2 or more developers. The developers’ collective method rationale is expressed in the added collective goal rationale association. The multiplicity 1..* indicates that there has to be a common goal for this association to occur. Subsequently, we must also declare an association has the possibility to describe how a group can experience a collective rationality resonance when they are considering of using a certain method component. This is declared in our adjusted diagram in a ternary association between the classes; goal, method component, and group. A UML Class diagram (Booch *et al.*, 1999) describing these added aspects of method rationale and the possibilities of achieving rationality resonance on an individual and a collective level is depicted in Figure 38.

![Figure 38 Rationality Resonance individually and in groups](image_url)

**8.6 Concluding Discussion**

The modelling seminar proved to have a lot of situations and discussions where method rationale was clearly present. During the analysis I discovered three communication characteristics which became my core categories;

- Method rationale as attention director
- Method rationale as consensus builder
Method rationale as check point

The communicative characteristics in my case study clearly showed how method rationale plays different roles in communicate about methods. I would go so far as to say that the presence of method rationale is vital for the communication in the described characteristics. It had a unidirectional function of directing attention as P explicated the method rationale of the various method components we used and explained what they were good for. He also directed attention to the rationale behind the task and helped the group to think about how this rationale would play with the one offered by the method components. At times he also directed attention towards the method rationale of various method elements when the group was trying to ascertain whether the method component would have to be complemented with additional concepts in order to capture the business of the Armed forces properly. When method rationale is used as an attention director, it serves the role of creating a mental space where rationality resonance can occur.

Multidirectional communication was found in the other two communication roles; method rationale as consensus builder and method rationale as check point. These two communication roles were classified as multidirectional since they involved statements between several group members. When method rationale served as a consensus builder the multidirectional communication involved discussions and arguments concerned how the group members agreed on something. Agreement was found to be reached when the group could realise that their rationality coincided with the method rationale offered by the method component being used. In the characteristics where method rationale served as check point it played a very important role as it actually was the rationalities of the group, the task and the method component that were compared to each other. The situation where these rationalities match can be likened to the state of rationality resonance, a state where the method is used and its’ proposed rationality coincides with the rationality of the method user (Russo and Stolterman, 1997). The findings from this case study indicate that the concept of rationality resonance could be extended into other situations besides individual method usage. Using a method is seldom a one man task. By extending the concept into also addressing issues on a collective group level, as I have done, would help researchers to better understand the interplay of various rationalities in a team and how decisions about analysis and design of information systems are taken. We would also be able to better understand the role method rationale play in situations where a method is communicated to novices, like in teaching situations. For that reason the UML class diagram describing method rationale in relation to rationality resonance introduced in chapter 4 was complemented with classes and associations capable of expressing rationality resonance on a collective group level. The result is described in Figure 38.
My conclusion is that method rationale is a fundamental part in communication about methods. It serves several purposes as described above, purposes that are fundamental to understand in order to understand the complex business of contemporary systems development.
9. Method Rationale and Method Components in the Context of Teaching Information Systems Development Methods

This chapter describes a case study where the explication of method rationale through the method component concept is tested in a practical context. The reason for this is to ascertain the viability of the developed concepts and theories by applying them in a situation where they could prove to be beneficial. My ambition in this chapter is to investigate the communicative potential of the method component concept in a very common situation, namely a university course in systems development. The goal is to highlight differences in how the systems development method has been internalised between two groups of students and to explore the possibilities of applying a method component perspective on systems development method training.

Fitzgerald (1997) describes method usage in relation to experience in Figure 39. When it comes to actual use of methods, empirical evidence (Fitzgerald, 1997) has been presented that shows that developer experience is clearly related to the extent to which methods are used. Inexperienced developers tend to use methods as they were presented to them during their education. As they gain experience they gain more confidence and are more willing to take chances and abandon a lot of the support formalised methods can offer, and thus method usage decreases. The empirical evidence also shows that more experienced developers later on tend to realise the problems connected with a systems development method that is completely informal. As a result, the developers start to appreciate the support that formalised methods can offer and thus method usage increases. However, this time around the methods are used in a more insightful fashion and tailored to the actual project needs. This individual experience process amongst developers is shown in Figure 39.

![Figure 39](image-url)
As suggested by Fitzgerald (1997; 1994), rationality resonance is more likely to occur after having used one or several methods for a long time—just as any knowledge, methods-in-use are part of an ongoing institutionalisation process whereby the knowledge is continuously integrated into our world and eventually becomes part of who we are as social beings (cf. Berger and Luckmann, 1967).

Research has shown that traditional education of system development methods is flawed and that the education and formal training of systems development methods ought to be altered as it is often focused on the process on following instructions rather than reflecting about what is actually being done (Mathiassen and Purao, 2002) or that traditional training in systems development methods use might result in shallow cognitive structures in future systems developers (Lee and Truex, 2000). The question is how a change in systems development method training can be done and what possibly could be achieved. Generally, students have a somewhat rigid view on systems development method usage in comparison to more mature users of systems development methods. According to Fitzgerald (1997), maturity means that the user of a systems development method uses the method because s/he wants to use it, because s/he finds that the method can provide useful, important, and needed support for various tasks during a systems development project. This situation can be likened to a state of rationality resonance where the public rationale provided by the method is in harmony with the private rationale experienced by the individual method user (Russo and Stolterman, 1997). That is, when the underlying philosophy of the method is truly embraced by the method user so that method prescriptions make intuitive sense.

The rationality inherent in methods and their use—a.k.a. method rationale—thus appears to be central to how method users learn to appreciate methods. One reason that extensive practical method experience is supposedly required in order for rationality resonance to occur, might be that method rationale is seldom explicitly dealt with in method textbooks, and thus not in traditional education and formal training.

The question, then, is if a better foundation for understanding methods can be laid already during the initial learning phase (as opposed to the lifelong in situ learning by doing) by explicating method rationale and, by doing so, training future developers in thinking about methods at a more abstract ‘philosophical’ level. Teaching this type of method knowledge would then prepare students (the future systems developers) for their use of methods and facilitate their lifelong learning process—giving them a deeper understanding of the practice of systems development and a solid foundation for becoming competent method users.

While previous research on method rationale (Ågerfalk and Fitzgerald, 2006; Rossi et al., 2004) points in this direction, no empirical work on the role of method rationale in teaching and learning has been reported in the literature. Therefore, the aim of this chapter is to explore the impact of explicating method rationale in method teaching on students’ ability to understand the suitability of
a particular method to a specific development setting and thus helping them to reach rationality resonance. The case study in this chapter can be placed in the method-in-action sphere in Figure 14.

9.1 Chapter Overview

This chapter is a report from an experiment conducted during the course of systems development methods at Örebro University. In relation to my overall research process defined in Figure 4, this chapter covers two activities. The first activity and, the main contents of the chapter are devoted to the *Empirical grounding Teaching Case* activity. During this activity, I am testing explication of method rationale through method components in a teaching context. Thus, inputs for this activity are the method rationale definition and the method component concept. The method rationale definition is also considered to be output as the definition might evolve during the case. The output is defined through the *Teaching results*.

The Teaching results are the primary input to the second activity of this chapter. In this activity I, once again, subject the method component concept to a conceptual re-modelling and analyse whether it needs any adjustments. Other inputs for this activity are the method component concept as it is the primary object of study. It is also defined as output since it might be changed during the re-modelling activity. Any changes in the method component concept would also involve adjustments of the initial design goals. Hence, these are defined as both input and output.

Possible alterations of the method rationale definition during the re-modelling activity must also be considered. Subsequently, this definition is regarded as both input and output from the re-modelling activity. Any discovered alteration of the method rationale definition is discussed in section 9.7.

A model depicting the chapter details is shown in Figure 40.

![Figure 40 Chapter 9 details](image)

More thorough descriptions of how specific data was captured and analysed will be found in sections 9.4 Lectures, and 9.5 Modelling Seminars. In these sections, I have described how data was captured during these above mentioned situations before I give an account for my experiences and interpretations and analyse my findings. The chapter ends with a conclusion where I sum up my experiences from the case study.
9.2 Case Setting

The case study was conducted during a university course in object oriented systems analysis and design. During this particular course students are taught two different systems development methods. The major part of the course is devoted to the study of Rational Unified Process (RUP) (Kruchten, 1999). The first four weeks of the course cover essential RUP elements. A selected portion of the RUP is taught through lectures. At the end of each week seminars are held where groups take the role of modelling leaders performing analysis and design. The pedagogical thought is that a mix between lectures and a more hands-on approach can give the students a good idea of what the systems development method is all about and an idea of how to use it. Lectures on more general systems development method knowledge are also interwoven between the method oriented lectures and the seminars. Existing research is presented and some theories on systems development are introduced in order to give the students a context and a frame of reference for the method oriented lectures.

The last week is concentrated on a systems development method (VIBA/SIMM (Ågerfalk, 2003)) with a slightly different perspective than RUP. The rationale behind this is to give the students an idea how perspectives are manifested in systems development methods. An easy way of distinguishing the two systems development methods from each other is to say that RUP has a traditional object oriented perspective and that the other, contrasting, systems development method is founded on theories on communicative action. VIBA/SIMM was chosen for this research because (a) it was already part of a regular university course, (b) the course included practical assignments, (c) VIBA/SIMM is explicitly grounded in an articulated theoretical framework—Actability theory (Goldkuhl and Ågerfalk, 2002), and (d) it is well suited to structure as a collection of method components.

For the sake of this research case the order was reversed and the students were taught VIBA/SIMM first. The reason for this decision was that I did not want them to become influenced from the lectures on RUP or the lectures on general systems development theory. Hence, roughly the first full week was devoted to concentrated study of VIBA/SIMM. The schedule for the VIBA/SIMM part was the following:

Week 1
Monday – Lecture on Workflow one (Two hours per group)
Tuesday – Lecture on Workflow two (Two hours per group)
Thursday – Lecture on Workflow three (Two hours per group)

Week 2
Monday – Seminar on VIBA/SIMM (Group 1, four hours)
Tuesday – Seminar on VIBA/SIMM (Group 2, four hours)
After the case study the course went back to the original curriculum and spent the rest of the four weeks of the course to RUP. The number of involved students was 24.

### 9.2.1 Maturity in the Group

The group could be considered novices. They did not have any formal training or experience in VIBA/SIMM before. Some of the students had encountered parts of it, since some components also were included in the method called Change Analysis/SIMM (Goldkuhl and Röstlinger, 1988) which some of the students had encountered.

The course was on intermediate level so when it came to experiences in other systems development methods, the students had only encountered fragments earlier in their education. Examples of this could be an approach to model a database with is introduced during the first semester. None of the students had any general or specific prior knowledge on systems development theory or research.

The students were aware that they were participating in a research project and that the lectures differed. Furthermore, they were asked to refrain from discussions across the group boundaries concerning the lectures and experiences during the course.

### 9.2.2 The Systems Development Method VIBA/SIMM

Versatile Information and Business Analysis or VIBA/SIMM (Ågerfalk, 2003) is a systems development method founded on theories of communicative action and actability. Founded on theory of speech acts, the systems development method VIBA stands for an alternative way of analysing requirements for various systems. Traditionally, a lot of systems development methods have been oriented towards description of reality through the use of models describing business information. These models are then used to inform others of what goes on in reality. The fundamental difference between VIBA and more traditional description oriented systems development methods is that VIBA takes the point of departure in an appreciation of the communication taking place by use of the system being developed. More concretely, this means that instead of the initial focusing on descriptive information, as emphasised in traditional systems development methods, VIBA starts off in an analysis of how the information is to be used in communication between various business actors.

This is a language-philosophical standpoint that emphasises that people act when they use language. This is done, for example, through promises, orders, threats and so on. Subsequently, communication through an information system also shares these characteristics. An update in a database concerning an object being removed from stock is not performed because the model needs to be updated to reflect reality, it is done because there is a need to communicate
knowledge concerning the removal. Someone needs to tell someone else that something has happened (Ågerfalk, 2003).

9.3 Interpretative Research Approach

Given that little research has been conducted in the area of method rationale and method teaching, an exploratory qualitative approach was deemed appropriate, as suggested by Miles and Huberman (1994). The case study was conducted with an interpretative research approach (Walsham, 1995) since my research question during the case study was to explore the communication potential of method rationale and the method component concept in a teaching situation. The exploratory approach undertaken focused on communication through actions taken, students’ behaviour and verbal accounts during lectures and modelling seminars, e.g., error rates, number of teacher interventions and the degree of activity during seminars. The ambition was to analyse and compare the differences between two student groups who had been taught the same systems development method, but in two different ways. Part of the test aimed to explore the possibilities of actually applying a method component perspective in a teaching context but that primary task during the case study was to ascertain differences in the student groups’ understanding of VIBA/SIMM.

When it comes to interpretative analysis it is important to be aware that the generated results are coming from the researchers’ subjective interpretations. In order to present a strong argument by help of interpretations it is important to give a rich description of the case and the analysis in order for the reader to be understand where the conclusions come from (Walsham, 1995; Geertz, 2000). As the object of study was communication of method rationale, this concept will be in the centre of attention in order to capture the relevant social dynamics in the case study (Dyer Jr and Wilkins, 1991). This means that I will focus on and describe statements and actions that can be considered being about method rationale according to my interpretations.

To facilitate analysis, the modelling seminars were video recorded. Specific descriptions of how the lectures and seminars were planned and conducted, from a research point of view, will be found under respective headings.

In chapter 2, there is a description of Eisenhardt’s (1989) distinctions between three types of theory use in case studies. They can serve as a way to facilitate an initial guide to design and data collection, as a phase in an iterative process of data collection and analysis and, last but not least, as the resulting research product. The concept of the method component, including its focus on method rationale, was used as a theoretical descriptive framework that guided data collection and analysis (Patton, 1990). Subsequently, this case study will follow the second use of theory according to Eisenhardt (1989) as the case is a way to put previous theoretical findings into action, and a willingness to let them evolve or change in an iterative fashion. How this is achieved in this chapter is described in Figure 40. There, the method rationale definition and the method
component concept are treated as a theoretical foundation for this case study. At the same time I also revisit the previous definitions and analyse the conceptually in the light of new findings, thus, allowing them to evolve and change in accordance with my chosen standpoint in relation to Eisenhardt (1989).

It is important to make a note on my role during this course and case study. I shifted from teacher to researcher from time to time. During the lectures I took a teacher perspective and followed my plan for each lecture, presenting VIBA/SIMM in two different ways as objectively as I could. During the seminar sessions, I tried to take the role of a researcher as much as possible since I wanted to evaluate the students’ performance. However, at times I had to return to a teacher role in situations when they needed guidance about what to do. Examples of these types of situations could be when none of the students had noticed errors or when they were in a state of confusion that hindered any progress. During the modelling seminars, I had a colleague taking notes on what was taking place in order for not missing important incidents when I sometimes had to take a teacher role. This ensured that it would be possible to maintain an interpretative focus through out the modelling seminar even at times when I was mentally in the role of the teacher.

I must also add a note on the ethical issues involved in the case study. The part of the course devoted to VIBA/SIMM is one week out of five. Normally, VIBA/SIMM would be presented last in the course as an alternative approach for systems development compared to the RUP, which constitutes the main parts of the course. As already stated, the presentation order was reversed in order to ensure that the RUP part would not influence the students. I wanted to have a demarcated section of the course where a case study could be conducted. By reversing the order I could also ensure that aspects that might not have been covered in the lectures given to one group could also be communicated afterwards. This means that I made references back to VIBA/SIMM during the RUP part of the course and explained the method rationale dimension of VIBA/SIMM differed from RUP. I also introduced the method component concept through the RUP and explained its relation to method rationale to the students.

9.4 Lectures
The students were randomly assigned to two test groups, henceforth referred to as G1 and G2. The G1 lectures focused on reflection and the method’s underlying rationale using the method component concept. The G2 lectures focused on the method’s prescriptive dimension. The G2 students were tutored to perceive the method as an instruction for how they were supposed to act. These lectures ignored any reflection and reasons for why the method was constructed the way it was. Both groups were given an equal number of lectures and equal time devoted to hands-on training. Since the case study aimed to explore the possibilities to teach novices into a more mature level of systems
development method use than normal, it was imperative to give the two groups equal time devoted to hands-on practical training and lectures on the principles of how to use VIBA/SIMM. The differences between the teachings of the two groups lied in the amount of theory explaining the underlying rationale of VIBA/SIMM and references to the method component concept.

9.4.1 G1 Lectures
The G1 students received a total of three times three hour lectures on VIBA/SIMM. The method was presented as a collection of connected method components fulfilling individual as well as process related goals. By giving the students this presentation of the systems development method they had a chance to understand the method as a network of goals. The idea was also that they would understand the individual benefit of each method component more easily since they would have an easier time understanding how and where the individual components give necessary input to other components and why the various method components would need particular input from others. This way of presenting a method to students differs from the more process oriented view that is more commonly in use in traditional education (Mathiassen and Purao, 2002). The traditional educational models seem to be more inclined to emphasise the steps one must follow to use the method than emphasising the reasons for those steps. Underlying theory may normally be covered in a methodology course as well as hands-on training, but the idea to present a systems development method from a goal oriented method component perspective is new. By doing this, students could possibly have had an easier task of to connecting the suggested method prescriptions to the underlying philosophy or theory since the explicit goals made the founding theoretical standpoints more clear and intelligible. The students could then see how the values and goals emphasised by the theoretical base manifested themselves all through the method description.

Figure 41 shows one of the network views of VIBA/SIMM that were presented to G1 during the course. It describes the primary document sets from workflow three where the theory of actability really comes into the focus. The smaller rectangles are document types and the arrows describe their relations in terms of offered output and suggested input. This network form of presentation makes it very clear that you may require an E-interaction list and a document prototype to produce a document statechart, at least they can be considered as recommended input.

For every method component, the inherent rationale was explicitly discussed and presented. References were made to actability theory (Ågerfalk, 2003) to explain the goals the individual method components were supposed to fulfil.

The G1 students were also given a document designed to facilitate internalisation of the method component concept. This document was given to them as an aid to create personal meaning of how each of the method
components could be used. The documents had a simple layout. It basically consisted of four rows and a larger “free” area where the students could jot down any notes, thoughts or mnemonics they felt they required for their internalisation process.

The four rows addressed the method component name, the overall goal of the method component, the recommended input artefacts, and references to other components where the offered output would be useful as input. The “free” area notes could be, for example, what the notation looked like, or general thoughts about the method component. The students were encouraged to use the note area in any way they liked as long as they used it in a way that would help them to grasp the method component more easily. A fitting analogy would be to view the method component documents as templates for method component knowledge. Working with the method component documents was easily grasped by the group and they asked many different kinds of questions, seeking feedback on what they had made notes of. Overall, there was a feeling of creativity as the students internalised the method by establishing the method component perspective through their own personalised notes. The value dimension was taken out of the method component document since it had been deemed as impractical in practical situations.

As the lectures went along, smaller practical problems were presented to the students to give them an opportunity to try out their newly acquired skills. During these situations the students tried to solve certain tasks and problems during a systems development project. The idea was to let them attain some practical experience of how VIBA/SIMM could be used. These practical tasks

Figure 41. Network describing a view of VIBA/SIMM
also served as checkpoints as it made it possible to get an idea of how much the students had concerning of the method. They were never intended to serve as the sole practical experience of VIBA/SIMM as a modelling seminar was planned to take place the week after. They were rather a way to let them try the systems development method as the course went along.

During the lectures the students were very active and asked a lot of questions. The questions varied from notation techniques to actability theory and I really got the feeling that they were genuinely trying to comprehend as much as possible. I also noted that the students got comfortable in the group as they seemed to get more courageous as the course went along. They did not hesitate to ask numerous questions and they were keen on giving comments and suggestions to each other in a constructive manner. Overall, the lectures had an air of creativity and high tolerance as the students worked with the method component documents and the practical tasks.

9.4.2 G2 Lectures

The second group, G2, did not receive any of the above. Instead the focus during the lectures on VIBA/SIMM was on the step-by-step actions taken when following the prescribed actions suggested by the method. A fitting analogy would be to view the method as a cook-book telling a chef how to prepare a certain meal. The students were only told how to perform the actions in order to drive a project further. They were not told any of the goals achieved when artefacts were produced or how the artefacts helped realise goals elsewhere in the development process. They did not receive any references to Actability theory explaining why VIBA/SIMM prescribed what it prescribed and what the methods’ creators tried to achieve.

This emphasis on strict process flow for G2 yields a picture of the systems development method as a totality, as a rigid whole as opposed to the emphasis for G1. The general idea was to implement a more traditional pedagogical situation where reasons why and underlying theory is not clearly present. An important note on the differences between G1 and G2 are that the lectures for G2 generally were a bit shorter due to the lack of explanations of the underlying perspective. It takes a lot less time telling someone to do something instead of telling someone to do something and at the same time also explaining the importance and rationale for what’s being done.

Both G1 and G2 received equal amount of practical tasks to solve during the lectures. I found it important that both groups had an opportunity to use the method in a way that would give them a reasonable time of practical training before the planned modelling seminar.

The method was presented in a very straightforward manner and the students were very focused during these lectures. Actually, the level of focus during the lectures given to G2 seemed to be higher than during the lectures given to G1. I make this observation based on the fact that the questions that did come up were
totally focused on the step-by-step process and that you could get the feeling that they were concentrating on following and understanding this process. However, this form of presentation resulted in fewer questions from the students as compared to the G1 lectures. My interpretation is that the G2 students regarded the method to be less problematic and easier to comprehend. They did, however, understand VIBA/SIMM as more much more rigid than they did in G1 as the method presented to them had the structure of a plan or schema for success rather than a useful toolbox of components.

9.5 Modelling Seminars

During week two G1 and G2 were asked to demonstrate their practical knowledge of VIBA/SIMM through actual method use in a fictitious case. The groups were given a text describing a business with enough content to start an analysis and design. Both groups were given four hours in total for the task. This may seem like a very short timeframe but the point was not to produce a complete solution, rather to show that they can use various parts of VIBA/SIMM and understand how the method works. In order to make sure that there would be a possibility to actually reach something that looked like an end result, the students were told to ‘move along’ when needed.

The main point with the modelling seminars was to let the students try to be responsible for and sit in the driver seat during a modelling activity. During these seminars, the teachers act as business stakeholders, answering any questions about the business the students may have. The student groups are divided into smaller groups of about four individuals who take turns leading the modelling seminars. The rest of the students take the role of analysts and designers giving the modelling leader input and support. The seminars are meant to simulate a real method usage situation with the notable difference that the business being analysed is represented by one or two individuals and the modelling team (real life consultants) are represented by around 15 persons.

The seminars went on for four hours for G1 and G2 respectively. The larger groups being divided into smaller ones taking turn leading the modelling seminar. During the seminars, notes were taken by an assisting researcher. His task was to log important events on what was going on, especially if things were in a state of confusion and nobody knew what was going on or how to move forward. At these times the lecturer often had to step out of character to give the needed directions. Notes were also taken describing positive things that took place. If students solved a particular problem in an especially competent way, notes were taken describing those situations. All notes were time stamped.

Two video cameras captured the events during the seminars. One camera was directed towards the whiteboard where the students documented their results. The whiteboard was a natural centre of attention during the modelling seminars since everything that was going on was documented there. The other camera was facing the other students not responsible for leading the seminar at the time.
This way we not only captured what was agreed upon and how the students performed, we also captured some of what was not agreed upon, typically conflicts or arguments.

For each of G1 and G2, eight hours of video material was captured. A very important reason for the use of video was that it enabled us to pick out segments describing the performance of the different smaller groups, as well as the larger groups (G1 and G2). This gave us the opportunity to show selected video clips to the student groups and ask them about their individual actions, or non actions. This was important as it gave a possibility to ask them about their own experiences and interpret whether or not they acted in a possible state of rationality resonance.

9.5.1 G1 Modelling Seminar

The modelling seminar with the G1 group went along without any major problems or confusions. Of course there were times when work halted due to some confusion but these problems were generally solved by the students themselves. I did not feel it necessary to make a lot of interventions since they tended to take care of driving the task forward by their own accord. However, I did intervene when errors were missed or when a state of unsolvable confusion arose. In total, the number of interventions counted 22 times during the four hour seminar.

The general atmosphere was positive and the students appeared to have good confidence in their task. A lot of laughter and jokes during the seminar vindicated this. Very often the groups wanted to anchor decisions by asking questions like “Do we agree?” or “Does anyone have any objections?” These types of questions tell us that they felt a natural ambition to reflect upon what had been done and what needed to be done—to create an inter-subjective understanding of the task, the method, and the situation.

The use of study materials (method textbook, personal notes, and method component documents) during the seminar was very limited. They were mostly used in situations when the students wanted to verify that they actually took the actions recommended by VIBA/SIMM or to double check notation issues.

On several occasions during the G1 seminar, the modelling leaders expressed statements such as “Perhaps we should go back and review what we have done earlier?” and “This is something we can pick up again later.” These statements indicate a presence of iterative thinking as the students realised that they can not fully comprehend and analyse every relevant aspect of the business case instantly. At the same time this did not mean that the students settled for a quick-and-dirty analysis. They rather accepted the fact that they only could, and needed to analyse the business enough to drive that task forward. Throughout the seminar, a lot of discussions arose about central concepts in the business case and how the various method components could be used to deliver the required results. These discussions were not confined to the modelling leaders.
but involved the whole classroom as all students took an active part in working with the task.

Notation issues never appeared to be problematic. The knowledge about how the results were to be documented according to VIBA/SIMM resulted in very few errors and other G1 members often corrected errors made. Consequently, the types of statements exemplified above and the discussions showed an iterative work pattern. Hence, the students displayed a fair understanding of how the various components in VIBA/SIMM could be used in relation to each other.

9.5.2 G2 Modelling Seminar

The G2 group perceived the modelling task to be more problematic and cumbersome. At times, the whole group was sitting silently, simply watching each other and hoping that someone could figure out what to do. At other times, the videotapes show large parts of the G2 group frantically flipping pages in the method textbook. They were looking for answers to what they should do or how to document a result they had agreed upon. The following student voices reflect a feeling of confusion: “OK… does anyone have a suggestion?”, “Is this really how you draw?” or “I need some help here!”.

Overall, the students in G2 seemed more lost than the ones in G1, which became evident during the modelling seminars. When the G1 students expressed confidence and unity, the students in G2 seemed to be hesitant and uncertain of what to do. The two groups clearly showed homogeneity with regards to these aspects. The general atmosphere of the G2 seminar was quite uncomfortable. At frequent occasions, the only active students were the drivers of the modelling activities. Moreover, often only one or two students in the driver group were in charge and actually said or did anything.

In situations where the work stopped completely it became necessary for me to intervene—just as during the G1 seminar. I chose to intervene when it became evident that they could not solve the problem at hand without help or when they made errors that would make it difficult or impossible to use the results later on in the modelling seminar. It should be noted that I did not make any interventions unless I was convinced that I had to. This is important to emphasise, since I make a point of the number of interventions. In total, I intervened at 49 occasions during this four hour session with the students in G2.

The iterative ideas that the method suggested were often forgotten or misunderstood. The students clearly had difficulties in understanding how and where they should use their previously conceived results later on in the process. As a result some of the tasks were performed twice.

9.5.3 Comparison Between the two Modelling Seminars

The general feeling during and after the two seminars was that the groups in G1 who had been taught VIBA/SIMM with an explicit rationale dimension performed better than G2. They made fewer errors, were less confused and needed the least teacher interventions to create progress. G2 on the other hand,
experienced more confusion, made more errors and needed a lot more teacher interventions.

By analysing the notes taken during the modelling seminars it is possible to sum up the number of errors, states of confusion and interventions for G1 and G2 respectively. Figure 42 shows a table with the results from the modelling seminars. The results clearly show that G1 performed better when it comes to instantiating VIBA/SIMM into a method-in-action. By this, I refer to their ability to use the method more efficiently than G2. Using terms of measure is always difficult. Efficient in this context means that the students in G1 had a more unproblematic seminar than their counter parts in G2. They made fewer errors and were not as confused as G2. Overall, G1 had a considerably smoother and more trouble free seminar than G2.

Figure 42 describes the number of errors, confusions and interventions during the two seminars. By error I refer to a situation where the students use the systems development method in a faulty way according to the original method description. An example of an error could be when a student connects two objects that cannot be connected. In this category, G1 scored 6 errors and G2 scored 9.

![Figure 42. Diagram showing the results from the modelling seminars](image)

States of confusion are defined as situations where the students are not able to carry on working for some reason. These reasons can vary from not knowing which actions to take to not knowing who is in charge of the modelling seminar. A typical reaction during confusion among the students was to start reading in the method description to see if it could provide any input what to do. G2 was considerably more often in a state on confusion. During the modelling seminar they ended up in a state of confusion a total of 48 times, in comparison to G1’s 14 times.
Interventions are situations were the teacher’s opinion is called upon or when the teacher must intervene and comment on what is going on in order to correct faulty actions or to resolve states of confusion. A typical situation of this latter type is when no one is talking and the students simply look at each other hoping that someone else knows how to move the task forward. That means that every intervention is preceded by either an error or a state of confusion. However, not every error or state of confusion required an intervention as seen in Figure 42. At times, the students were able to rectify their errors or find a solution for how to solve a state of confusion. G1 needed an intervention at 19 occasions. The corresponding result for G2 was 49.

Generally, the groups were more active and talkative in G1. Most of the students tried to actively take part in the modelling seminar and the process of driving the seminar further went a lot smoother when compared to G2. G2 experienced a lot more situations of confusion and needed a lot more interventions to make progress. This often manifested itself during the modelling seminar in situations where the students sat quiet, reading the VIBA/SIMM manual to get an idea what to do next.

9.6 Method Rationale Lessons Learned

One important aspect of this case study is the fact that method rationale as a concept has been introduced to only half of the students. This raises a need to analyse the students’ behaviour during the lectures and modelling seminars from a perspective that does not imply that they have to use my defined terminology to describe things in a way that I would consider insightful.

This case study has rather been an attempt to communicate a systems development method to a group of students in two different ways and subsequently analysing and comparing their performance. This leads to certain problems related to measurement. How is it possible to measure performance among a group of students working together? In my case I have presented statistics from the modelling seminars that imply that one G1 performed better than G2. I choose to interpret these results in the light of rationality resonance (Russo and Stolterman, 1997). In Figure 38, presented in chapter 8, I present a conceptual model of rationality resonance in groups. In this case study I once again return to the notion of collective method rationale and the possibilities to achieve rationality resonance on a collective group level. I have therefore chosen to treat the two student groups as separated objects, each group has a possibility to achieve a state of collective rationality resonance.

The statistical results are interpreted as incidents where rationality resonance was clearly absent. Since rationality resonance is a state where the public rationale of a method is used in harmony with private rationality, I interpret the incidents of rationality resonance absence as a level of internalisation. A high level of internalisation would result in fewer incidents where rationality resonance would be absent. On the other hand, a lot of incidents would lead to a
conclusion that the level of internalisation is lower. By comparing the two groups in this light I have a possibility to present an opinion about the two groups’ possibilities to achieve rationality resonance. My results from the case study indicate that communication of a systems development method with a method component perspective, explicating its method rationale, has a positive effect on the students’ possibilities to internalise the method. The case study is also an example of EME in a method-in-action sphere setting as described in Figure 14.

When it comes to further evolution of the understanding of method rationale as it was understood in chapter 8, I have not found any requirements for change during this case study. Thus, I leave my diagram expressing method rationale and rationality resonance unchanged.

9.7 Conceptual Re-Modelling of the Method Component Concept

The method component concept was implemented in this case study through the use of method component documents. The role of these was to help the students to perceive VIBA/SIMM as constituted of a number of individual method components. Besides expressing the goal of each method component the method component documents also described their interfaces, explaining how the method components were related to other method components. Thus, we returned to the artefact types defined as prerequisites (input) and deliverables (output) and applied them in practise.

The concept of values in relation to a method components goal was left out for the same reasons as described in section 7.4. They are conceptually correct to incorporate in the definition. However, they are of less practical use in many situations.
Apart from implementing the method component concept in method component documents, no other alterations of the concept and its definition was deemed necessary. Thus, also this definition remains unaltered.

**Figure 44** Method Component Concept unchanged

### 9.8 Concluding Discussion

The purpose of this chapter was to explore the impact of explicating method rationale in method teaching on students’ ability to internalise a systems development method and use it in a specific development setting. The concept of method rationale was operationalised through the method component concept during teaching.

To explore the potential of achieving rationality resonance I split the class in two groups: G1 and G2. The G1 students received teaching using the method component concept. They also received a theoretical background which explains the underlying reasons behind the suggested activities in the method. I also employed a method component document in G1 which aims to help the students to internalise a systems development method as constituted of individual components.

The G2 students received a more traditional presentation of VIBA/SIMM, the method in question. The lectures given to G2 focused the step-by-step process of VIBA/SIMM and omitted the underlying method rationale aspect and a modularised component view of the method.
When analysing the lectures and the students’ performance during the modelling seminars I can conclude that the G2 students had a shallower understanding of VIBA/SIMM than the G1 students. G1 students made fewer errors, required fewer teacher interventions and showed a higher degree of activity overall. This conclusion is founded an interpretation that the students’ level of internalisation is related to their performance during the modelling seminar. The level of performance is defined by the number of incidents where there have been errors, confusions, and interventions. These incidents are treated as states where rationality resonance is not present. Hence, by summing up incidents we can compare the two groups and make conclusions about their performance. Of course, a low number of incidents is interpreted as a higher level of internalisation.

Altogether I find support for the idea that explicating method rationale in communication of systems development methods may have a positive impact on students’ abilities to internalise and use methods. Method communication with an emphasis on method rationale may better prepare the students for their professional life as they reach a deeper understanding of how methods convey knowledge and how they can combine that knowledge with their own rationality. This would yield systems developers more capable in the complex business of systems development. A component perspective and the use of method component documents can be added to a curriculum to help the students to develop a deeper understanding of systems development methods and appreciate their underlying rationality.

Although the empirical validation reported in this chapter is limited to VIBA/SIMM, there is nothing to suggest that the method component concept would not also work in other communication contexts. In fact, the possibility to map the construct onto any existing method was one of the design goals of method component from the outset.
PART IV – Findings and Conclusions

*The most exciting phrase to hear in science, the one that heralds new discoveries, is not "Eureka!" but "That's funny..."*  
Isaac Asimov
10. Summary of Findings and Conclusions

This final chapter gives a summary of the relevant findings and constitutes the conclusions in my dissertation. It covers the theoretical chapters in PART II as well as the results from my three empirical case studies in PART III. In this chapter I will also return to my research question and answer it. I also return to the two fields of ISD and ME and include a discussion where I explain how EME can be used to overcome the weaknesses I have identified in the two fields. The chapter ends with a discussion about future challenges research on systems development methods will have to face in the future and explain implications for the field of EME.

10.1 Chapter Overview

This chapter covers my final activity, Concluding Analysis. During this activity I summarise my findings and the evolvement of the method rationale definition and the method component concept. Together, with these inputs I also return to my elicited design goals and the results from the method reconstruction case in chapter 7 and the teaching results from chapter 9.

The research question is also a required input for my concluding analysis. It will serve as a starting point for the formulation of the final output in my dissertation, the conclusions.

10.2 Summary of Theoretical Results

The dissertation is comprised of four primary parts. PART II is purely theoretical and addresses issues concerning how the field of method engineering can evolve in a way that would facilitate knowledge transfer to and from the field of ISD. PART II also investigates and defines the concept of method rationale and shows how this concept can play a function in an attempt meet the challenges proposed described in chapter 1. I show how this could be possible by synthesising a new ontological framework for understanding method rationale in the context of systems development methods by combining the respective strengths of the two fields of ME and ISD. This new way of understanding systems development methods field would yield an arena where
method rationale could play a central part as it would be equally present and
central in all spheres of method research. I chose to call this arena *Extended
Method Engineering*.

PART III contains one important theoretical contribution in the formulation of
a re-designed method component concept.

### 10.2.1 A Definition of Method Rationale

The most central concept in this dissertation is the concept of method rationale.
It is a concept that I have chosen to define with the help of UML (Booch *et al.*, 1999) as depicted below. The concept has functioned as input for the
interpreative analyses during my case studies. At the same time it is also a
concept that has evolved as I have learned more about how it can be interpreted.
My final understanding of the concept and its UML definition can be found in
Figure 46.

![Method rationale and rationality resonance individually and collectively](image)

The figure describes the concept of method rationale as *a relation between a
method component, the goal it realises and the values that motivate why the
goals can be considered desirable*. In the same way, a systems developer can
have goals s/he wants to realise and a value rationale structure that motivates the
goals. When the goal of the method component coincides with the goal of the
systems developer in a usage situation, referred to as method-in-action, we can
speak of a state of rationality resonance. During rationality resonance the two
rationalities (method goal rationale and developer goal rationale) are in harmony
and the method is used in a well informed manner.

When several method users are working together, in for instance a modelling
seminar, we can speak of a collective goal rationale. Of course, this refers to
situations where the group has jointly agreed upon that they indeed have the same goal. Hence, we can also talk about a collective rationality resonance.

### 10.2.2 The Method Component Concept

The dissertation also covers a design of a systems development method concept that has a foundation in method rationale. The concept has its origins in the method component concept (Röstlinger and Goldkuhl, 1994) and can be considered an evolution of the original concept. The motivation for creating a systems development method module concept was to create possibilities to focus on parts of systems development methods without losing sight of the method rationale dimension. The evolution was achieved through a design science process where we used conceptual modelling to yield a method component concept that would meet the requirements set by our design goals. The design goals were the result of an analysis in chapter 4:

**Self-contained**: It must be possible to treat the method component as a self-contained part of a systems development method with regard to the guidelines that describe the deliverable and the process of producing such a deliverable. This way a method component can be modelled out from a systems development method or act as a building block when constructing new methods or during method integration.

**Internal consistence and coherence**: The method component must be perceivable as an internally consistent and coherent entity. Another way to put it is to say that the construct should be stable over time and without lose ends. As a result the method component will be perceivable as meaningful. The principle is an ambition to create congruence amongst method components and homogeneity in systems development methods.

**Rationality**: Actions are performed and prescribed for reasons. Thus a method component must have an identified target state, a purpose for its existence. The arguments are motivated by the components inherent method rationale and address the goals and values that the method component realises.

**Connectivity**: The method components must be possible to connect with each other. Each method component, viewed as part of the systems development method, should contribute to a chain of goal achievements adding to the overall goal of the specific project. Thus method components must have the capability to express how they are connected to each other and how these connections can contribute to the overall goals linked to a specific project.

**Applicability**: It should be possible to map the method component construct onto any existing systems development method in any situation.

The design goals were the starting point for a process of designing a new systems development concept for use in EME. On aspect that had to be decided
was the appropriate level of granularity for the method components. Ultimately, a level of granularity that corresponded to the diagram level according to Brinkkemper et al. (1999) was chosen. Creating a modularisation concept is a very common task within ME. I had the ambition to create a method module concept that would be applicable in all three spheres of EME. This endeavour resulted in a re-designed method component concept capable of expressing method rationale. The definition of the method component concept has been allowed to change and reflect our findings during the case studies. The final version of the method component concept is depicted below in Figure 47. Again, we chose to describe theoretical concepts with the help of UML (Booch et al., 1999).

![Figure 47 The Method Component Concept](image)

The figure above shows the method component concept contents. This refers to the internal view where the elements that constitute the method component are exposed. The method component concept contain method elements of five different types; Concepts – are what is focused or talked about in the method component, Notation – is the way that results are recorded, Action – is the stepwise instruction telling a method user how to focus the correct concepts and record them, Artefact – is the main deliverable from a method component and defines its overall goal. By using a particular method component and producing its suggested artefact, one could realise the overall goal. The last method element is Actor role – it is a description of roles involved in the method component. An actor role can be associated to actions by being the role...
responsible for carrying out the action. It might also be the case that an actor role is participating in an action as an end user or customer. The figure also shows that each method element has a foundation in method rationale as there are defined associations to goals and their values. Furthermore, the overall goal of the method component is defined in an overall goal association.

Thus, a systems development method can be understood as an aggregate of connected method components, structured in a goal achievement hierarchy as shown above in Figure 48. As shown, one must also be aware that there might be goal contradictions.

10.2.3 The Field of Extended Method Engineering

As a way to overcome the problems defined in the existing research agendas about systems development methods, chapter 4 presents an alternative ontological framework for Extended Method Engineering, EME. It is founded in an analysis of the respective strengths and weaknesses identified in the existing fields of ME and ISD and consist of three spheres that together constitute an integration of the two fields of research.

Since the field of ME is naturally strongest in the method engineering sphere the top sphere will keep this name. The ME sphere has been reshaped to reflect what method construction rationale is. In reality, very little differ between the notion of method construction rationale and the notion of public rationale. The result is a combination into a public method construction rationale. Both aspects are equally important for pinpointing what constitutes a method, separated from any usage situations. This new category contains the philosophical paradigms covered in the field of ISD as well as the underlying goals and values.

This is the foundation for creating and defining ideal typical meta models of systems development methods. These are non-situated descriptions of a methods content and structure and can be used as a means to represent, construct or adapt systems development methods. The process of gathering requirements for adaptation of ideal typical meta models into situational methods is also covered in the method engineering sphere. An important difference is that the input to
the requirements does not only come from the development situation, but also from the actors’ private rationality as it is understood in the third sphere.

The second sphere bridges the problem of traditional ME not being able to focus systems development methods’ action dimension. In ISD this is covered by the notion of method-in-action. Subsequently, since ISD is very successful in explaining how systems development methods are actually used, the second sphere will be called the method-in-action sphere. In this sphere an EME researcher can focus on how a systems development method is used as a method-in-action. The method-in-action is understood as actors (method users) applying a method in social action as a means to realise the goals of the method as they are defined in the ideal typical meta model. This is also the sphere where rationality resonance can be found and studied. The EME framework also shows that the method user is influenced by his/her private rationality which resides in the method reflection sphere as goals and values are used in a hermeneutic process of interplay between the method-in-action sphere and the method reflection sphere.

The third sphere is the method reflection sphere. It is essentially the same in both original fields, but with a difference in focus. In ME the reflection basically concerns method use rationale whilst ISD research concerns the actors’ private rationality. The original elements have been combined, renamed and synthesised into new elements describing how the concept of method rationale could be focused in EME.

<table>
<thead>
<tr>
<th>Method engineering sphere</th>
<th>Method-in-action sphere</th>
<th>Method reflection sphere</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Situation</td>
<td>Method-in-action</td>
<td>Private rationality</td>
</tr>
<tr>
<td>Method use rationale</td>
<td>Carried out by</td>
<td>Intentions, Goals, Values</td>
</tr>
<tr>
<td>Arguments for redesign</td>
<td>Must</td>
<td>Human bias, Experience, Skill</td>
</tr>
<tr>
<td>Public method construction rationale</td>
<td>Explains</td>
<td>Indoor bias, Experience, Skill</td>
</tr>
<tr>
<td>Philosophical paradigms</td>
<td>Rationalises</td>
<td>Indoor bias</td>
</tr>
<tr>
<td>Goals</td>
<td>Ideal typical meta model</td>
<td>Indoor bias</td>
</tr>
<tr>
<td>Values</td>
<td>Adapted into</td>
<td>Indoor bias</td>
</tr>
<tr>
<td>Design decisions</td>
<td>Situational Method</td>
<td>Indoor bias</td>
</tr>
<tr>
<td></td>
<td>Rationalises</td>
<td>Indoor bias</td>
</tr>
<tr>
<td></td>
<td>Arguments for redesign</td>
<td>Indoor bias</td>
</tr>
<tr>
<td></td>
<td>Methods engineering</td>
<td>Indoor bias</td>
</tr>
<tr>
<td></td>
<td>Method-in-action</td>
<td>Indoor bias</td>
</tr>
</tbody>
</table>

**Figure 49.** Extended Method Engineering – a Framework describing how method rationale can be used in all method spheres
The third method reflection sphere is divided into two parts, one part that can be communicated easily and one part that is more difficult to fully express. This is done by appreciation of the *private rationality* dimension as expressed by the field of ISD with one important clarification. In essence, intentions are reasons for actions and are thus goal oriented. Goals do not exist by themselves. The describe target states which someone has deemed valuable for some reason. Thus, the intention element is divided into and defined as a combination of goals and values.

Outside the private rationality we find the *method use rationale*. In ME this was defined as captured arguments from method usage situations. According to ISD, method use rationale should rather be connected to and be considered a deliverable from the private rationality since the private rationality captures any reflections or experience from method usage situations already when it happens, long before any computerised tool can. Thus, the method use rationale is tighter coupled to human aspects than before. This method use rationale can then be used as arguments for redesign in the method engineering sphere. As a consequence this model provides for the possibilities of achieving double feedback. It does not only capture the feedback interplay between the actor and his/her private rationality. It also covers the feedback from method use situations into arguments for redesign of ideal typical methods. From a Weberian standpoint, the model focuses goal rational thinking equally through the three spheres.

10.3 Summary of Empirical Results

The dissertation contains four case studies in PART III where the definition of method rationale has been used as analytical input. The method component concept has been applied in two of the cases. The following summarises the case studies and highlight the relevant findings.

All case studies can be considered examples of EME research.

10.3.1 The Posten IT: E-Gov Case Study

Going back to the model describing EME in Figure 49, the E-Gov case study mostly resided in the method-in-action and the method reflection sphere, although most of it could be considered to belong in the method-in-action sphere. The case study covered the initial stages of a method implementation project and shows how method rationale plays a vital role when addressing the tasks connected to preparing the development organisation to change and to assess the current status of the project.

An organisation is typically reluctant to change and attitudes towards any organisational change might stir up conflicts. *Knowledge and understanding of this organisational inertia is created through communication about the project where method rationale is used to point out what can be gained by the change. An organisation must find a balance between the public method rationale the*
base method offers and the needs expressed through private rationality coming from the development teams. Otherwise, the development teams might end up in a state where they ‘awfulise’ (Davies, 2008) the implementation project where the affected developers tend to imagine worst-case scenarios.

Communication of method rationale during method implementation enables the affected persons to compare the public method rationale of the method being implemented, with their own private rationalities. It gives them the opportunity to assess their possibilities to reach rationality resonance once the method implementation project is finished. Implementing a systems development method in an organisation means that method rationale is communicated back and forth between the people involved and creates meaning and sense.

The persons responsible for the implementation project have the roles of communicators of public method construction rationale during the implementation project. This involves finding pilot projects, success stories and carefully selecting method components to be implemented in the right place at the right time.

An important lesson during the Posten IT: E-Gov case study was that the method reconstruction phase possibly could lead to biased decisions. Some aspects of what the consultants did indicated that they might have already had decided what to implement and in what order.

This led to requirements for an improved method reconstruction process.

10.3.2 The Posten IT: PSU Case Study

This case study gave us an opportunity to test the method component concept in a method reconstruction task. This case fits best in the method reflection sphere of EME since it covered activities where we as researchers were helping method user to reflect on their own development processes. Going back to Figure 49, the reconstructed systems development method of PSU is also possible to use as feedback to the method engineering sphere.

In addition to applying a method component perspective on the implicit systems development method we also present an approach for method reconstruction as a series of steps.

Step 1 “Apply a method component perspective on the implicit process” A method component’s level of granularity is defined to correspond to singular deliverable artefacts. Thus, identifying all artefacts means that all method components are accounted for.

Step 2 “Ascertain what method rationale explains why they are doing things the way they are” This step analyses how the method components are related to each other through connecting method components to each other. It is done by analysing the artefacts role to other method components viewing them as either prerequisites or deliverables. The individual goals of each method component are understood as the goals reached by producing the deliverable artefact. Having defined a goal relational structure between the components enables the
person responsible for the reconstruction process to communicate the development method to other people. This process of reconstruction is conducted iteratively together with representatives from the project team.

Two additional steps concern an analysis of how the reconstructed development method compares to the method being implemented.

Step 3 “Compare the process to the method rationale offered by the method being implemented” This step contains comparisons between how method components helped to contribute to the overall goal of a development project.

Step 4 “If needed, make adjustments by adding or removing method components” In our reconstruction case study, we never performed step 4 as the consultants deemed it unnecessary to enforce a RUP perspective on an otherwise well function project team.

Compared to the approach for method reconstruction in the E-Gov project assessment this approach yields a description of the development process that could be communicated to others more easily since it would result in a diagram consisting of method components that are connected to each other through goal rationale. Our results indicate that the assessment phase can benefit from applying a method component perspective during tasks that involve reconstruction of an existing implicit development process. If the development process is implicit, as was the case with the PSU project, the method component concept can help to create a reconstruction of the involved artefacts and communicate their roles in the project. The suggested approach allows comparisons with the systems development method considered for implementation and helps method implementers to an enhanced understanding of, and a possibility to communicate about, the development situation.

Our experiences during the method reconstruction case study made us realise that the practical use of analysing the value anchoring structure of a methods rationale might be of lesser practical use, even though it is possible to identify situations where analyses of this type might be useful or even necessary. It is important to note that even though the value dimension might be of less importance in practical situations, it is very important in the theoretical understanding of method rationale.

10.3.3 The Swedish Armed Forces Case Study

This case study focused on method rationale during a five day modelling seminar at the Swedish Armed Forces. The task of the modelling seminar was to ascertain if the RUP could be used in a way that would enable the Swedish Armed Forces to describe their operations in a way that would not only capture military codes and regulations, but also be possible to use for future development of military systems. The case study aimed at identifying communicative roles method rationale can play during method-in-action. Three distinct roles were identified:
Method rationale as attention director – This role is a unidirectional function of directing attention as someone is explicating the method rationale various method components and explaining what goals they fulfil. Attention can also be directed to the rationale of the task and helps a group to think about how this rationale would play with the one offered by the method components. When method rationale is used as an attention director, it serves the role of creating a mental space where rationality resonance can occur.

Method rationale as consensus builder – This role exemplifies a multidirectional communication as it involves communication between several group members. When method rationale served as a consensus builder the multidirectional communication involved discussions and arguments concerned how the group members agreed on something. Agreement was found to be reached when the group could realise that their rationality coincided with the method rationale offered by the method component being used.

Method rationale as check point – The method rationale as check point is also considered to be an example of multidirectional communication as it involves how a group collectively compares the collective group rationale with the used method component and the elicited results. When a group is evaluating their results, they are essentially evaluating whether they were able to reach rationality resonance or not.

The findings from this case study indicate that the concept of rationality resonance could be extended into other situations besides individual method usage. Extending the concept into also addressing issues on a collective group level creates possibilities for researchers to better understand the interplay of various rationalities in a systems development team and how decisions about analysis and design of information systems are taken.

Method rationale plays a fundamental role in communication about systems development methods, not only when a method is taught to someone who has to internalise it. It also serves in various communicative roles in creating meaning concerning systems development method-in-action.

10.3.4 The University Course Case Study

The purpose of this case study was to explore the impact of explicating method rationale in a method teaching situation. The ambition was to measure differences in the level of internalisation between two groups of students in a university course. Half of the students in the course (G1) had been presented a systems development method (VIBA/SIMM) with an explicit focus on the method’s underlying rationale and with a method component perspective. G1 also were introduced to a method component document which aimed to help the students to internalise a systems development method as constituted of individual components.
The other half of the students in the course (G2) received lectures with a perspective on the process and the stepwise tasks the systems development method suggests. These latter students were not introduced to the concept of method rationale or method components during any of their lectures.

When analysing the lectures and the students’ performance during the modelling seminars I concluded that the G2 students had a shallower understanding of VIBA/SIMM than the G1 students. G1 students made fewer errors, required fewer teacher interventions and showed a higher degree of activity overall. This conclusion was founded an interpretation that the students’ level of internalisation was related to their performance during the modelling seminar. The level of performance was defined by the number of incidents where there were errors, confusions, or interventions. These incidents were interpreted as states where rationality resonance was not present. Hence, by summing up incidents we could compare the two groups and make conclusions about their performance. A low number of incidents were interpreted as a higher level of internalisation.

Altogether I found support for the idea that explicating method rationale in communication of systems development methods may have a positive impact on students’ abilities to internalise and use methods.

A component perspective and the use of method component documents can be added to a curriculum and applied to any systems development method to help students develop a deeper understanding of the method’s underlying rationality.

10.3 A Return to the Research Questions

In chapter 1 I presented my initial research question which this dissertation aims to answer. The question was formulated as follows:

How can method rationale be defined, represented and used in extended method engineering to improve communication about systems development methods?

This is the part of the dissertation where I return to my research question and present my concluding answer. The answer to the question is quite extensive as it basically involves all theoretical and empirical results I have presented so far in this dissertation and this chapter. The part that concerns the definition of method rationale can be found in section 10.2.1. Method rationale as it is defined can be used to understand as the underlying foundation (goals and values) that motivates the method appearance. The definition also encompasses the rationale of the method user and how method rationality resonance can be can be considered a harmonisation of the goals of the method and the method user. The definition also covers method rationale and rationality resonance on a collective group level.
When method rationale is represented in EME it should be represented in a way that shows a tight coupling between the method and its rationale dimension. This is important as it is difficult, and possibly pointless, to speak of method rationale without considering its relations to the prescriptive method counterpart. This coupling should be clear and defined to a level that makes it possible to speak of independent method components with explicit rationale dimensions. Hence, a method component concept is suggested as it can be considered to be the smallest meaningful method module, small enough to be modelled out of a method but still having enough content to be perceived as supporting method usage. The method component must have a prescriptive content (specified actions and corresponding concepts and a defined notation) as well a rationale dimension that motivates the contents appearance. How this is achieved is explained in section 10.2.2.

Using the method rationale definition and its corresponding method component concept in EME research has been tried in my case studies. These are examples of research typically belonging in the field of ISD but this time conducted with a foundation in ME concepts. Examples of this type of research can be found in section 10.3. It involves application of the method rationale definition as an analytical tool during qualitative case studies and application of the method component concepts as a way to interpret a method usage situation, and as a way to communicate systems development knowledge. Essentially all method application in EME can be considered communication. Methods are used to establish how communication is supposed to take place in a business. The results from method usage is recorded and used as a way to communicate the results to other people involved in, or affected by, the systems development process. Using a systems development method-in-action is to communicate with involved persons and using the systems development method as guidance for how the communication is carried out and concern. The method rationale definition and the method component concept are theoretical building blocks in the foundation for suggesting a new theoretical framework for systems development method research. In EME these concepts are used as communicative bridges to connect the previously separated spheres of method engineering, method-in-action, and method reflection. The theoretical concepts explain how the spheres are connected and enable communication between them.

10.4 Extended Method Engineering – A Synthesis
An extension of the field of ME has been suggested in this dissertation as a way to overcome the weaknesses in how systems development methods have been treated previously. The field of traditional ME has been extended to incorporate aspects that previously only have been covered in the field of ISD in an attempt to meet the challenges presented in the discrepancies between the two fields of ISD and ME.
In chapter 4, I present an analysis of strengths and weaknesses in the existing fields of ISD and ME. They propose a challenge for systems development method research and practice and can be understood as discrepancies between the fields and a need to overcome the negative aspects by incorporating the positive benefits from each field. The fields were analysed and the primary differences, in terms of strengths and weaknesses, were pointed out. The analysis yielded a model describing the strengths and weaknesses as shown below:

![Diagram showing strengths and weaknesses of ISD and ME]

This section aims to explain how the field of EME handles the weaknesses in the two fields at the same time as it takes the positive aspects into consideration.

The field of ME’s primary problems concern over-formalisation, a lacking perspective dimension, and absolutely no attention to the methods’ action dimension. The actual usage situation was not considered at all, even though much pointed in the direction that methods were not being used as intended even if the method was uniquely conceived for a project. Thus, we must have a method concept capable to apply in the understanding of actual method usage. This is handled in the framework for EME in the method-in-action sphere where this type of research can be applied with a higher degree of detail through the formalised method component concept and a defined model of method rationale. The method component concept suggests a degree of formalisation that is not too complicated but still detailed enough for traditional ME research. Thus, by bringing this concept into the method-in-action sphere as I have done in my case studies, shows that this type of research is possible. The method component concept has been defined in a way that makes it possible to tighten the relation between the method concept itself and its underlying perspective dimension through the incorporation of goals and values for the method components. In
this way we solve the problem with the field of ME having no perspective dimension.

The field of ISD had problems with a low degree of formalisation, no method communication principles, and no method management know-how. All these aspects are handled with the introduction of the method component concept. It allows a higher degree of formalisation and this makes it possible to store and manipulate systems development methods in a way that traditional ISD research has not been able to do, but still paying attention to the perspective dimension emphasised by ISD traditionalists. This means that EME utilises the ME field’s strengths in method management know-how as the formalisation and rigour of the method component concept makes it possible to keep methods and method parts in systems based on the UML definitions specified in this dissertation. A formalised method component concept also defines what can be communicated regarding methods as it specifies content and underlying method rationale. This makes it possible to emphasise what can be communicated regarding systems development methods. Actual proficiency in using a method comes with experience from method usage, but the method component concept would allow method knowledge to be communicated more clearly and explicitly, and hopefully allowing method users to easier become proficient more easily.

The framework for EME takes all the strengths and weaknesses of the respective fields of ME and ISD into consideration. Thus, I can claim that I have indeed synthesised the two fields in the suggestion of EME. The basic idea of a synthesis has had its foundation in enriching and extending the field of ME with the strengths from the field of ISD. I feel that this idea has been fulfilled through the field of EME.

10.5 Future Research on Method Rationale and Method Components

Of course, future research on the topics of method rationale and method components can be imagined. Traditional ME with this foundation has already been applied and tested successfully (Karlsson, 2005). The main challenges for the field of EME lie in further studies that explicitly use the formalised concepts of method rationale and method components in qualitative and interpretative settings.

Further research should be conducted in exploring the potential of these concepts during method training and general method communication. One might consider a systems development method manual that emphasise the method rationale dimension and presents the method as a chain of goal achievements though connected method components.

One can also imagine further studies that focus on different roles that method rationale can play in other settings than method usage. The role of method rationale during method construction might be such a setting.
References


Polanyi M (1983) The Tacit Dimension, Peter Smith, Gloucester, MA.


Electronic sources
Rational Software, The Rational Unified Process Version 2003.06.00.65
Publications in the series Örebro Studies in Informatics
