Integrating the rational unified process and participatory design for development of socio-technical systems: A user participative approach

Sofie Pilemalm, Per-Ola Lindell, Niklas Hallberg and Henrik Eriksson

Original Publication:

Sofie Pilemalm, Per-Ola Lindell, Niklas Hallberg and Henrik Eriksson, Integrating the rational unified process and participatory design for development of socio-technical systems: A user participative approach, 2007, Design Studies, (28), 3, 263-288.
http://dx.doi.org/10.1016/j.destud.2007.02.009

Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach

Sofie Pilemalm, Per-Ola Lindell, Niklas Hallberg and *Henrik Eriksson*, Department of Systems Development and IT Security, Division of Command and Control Systems, Swedish Defence Research Agency, FOI, PO Box 1165, SE-581 11 Linköping, Sweden

This study presents the MOPT-Systems Development Process, aimed at bridging the gap between ideality and reality. The process is based on an approach to systems development involving a formalised process for developing socio-technical systems. In specific, it integrates a modified Rational Unified Process (RUP) framework with a socio-technical system view and an extended participatory design (PD) perspective using PD techniques and social research methods. It is argued that the integrated approach, by combining the RUP formalisation, modeling tools and coverage of the entire development process, together with the parallel development of methodology, organisation, and personnel, will greatly enhance the chance of solid systems, grounded in the organisation and appreciated by its users. In this respect, the close cooperation with the end-users throughout the development process is supposed to contribute.

Keywords: Rational Unified Process, systems design, user participation, collaborative design

In the last few decades, the need for taking a simultaneous view on methodological, organisational, personnel, and technical aspects when developing information systems, i.e., to develop socio-technical systems, has become increasingly recognised (Avison and Fitzgerald, 1995). Similarly, the necessity of involving the end-users actively throughout the development process in order to arrive at systems that are actually usable, used and appreciated is today acknowledged by most system developers, at least in theory and in academia. However, simultaneously to these insights, the software engineering approaches that still dominate industry tend less to put explicit emphasis on the end-users and on the organisational and social aspects of information systems. An example is the *Rational Unified Process* (RUP) which has, in recent years, received much attention as a defined process for development of software intensive systems ensuring a high quality product (Kruchten,

Corresponding author:
S. Pilemalm
sofie@foi.se

www.elsevier.com/locate/destud
0142-694X $ - see front matter *Design Studies* ■■ (2007) ■■–■■
doi:10.1016/j.destud.2007.02.009

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

2004). On the other hand, socio-technical oriented approaches, such as *Participatory Design* (PD), are often criticised for being imprecise and lack in defining a fully specified design process (Constantine and Lockwood, 2002) and to put emphasis only on the early systems development phases resulting in that a ready-to-use system is seldom delivered (Tollmar, 2001). In the specific case of PD, the approach is almost exclusively applied to small-scale projects within the academic domain rather than to the design of large, strategic information systems (Oostveen and van den Besselar, 2004). In conclusion, the need to combine the benefits of a socio-technical perspective and active user participation with more formalised processes covering entire system life cycles seems urgent, both for academia and industry. This study presents a novel approach to systems development based on such a combination, for the development of information systems that are technologically solid as well as organisationally compatible and grounded in users' needs.

1 Study objectives

The objective of the study is to present an overall approach and a specified process for developing socio-technical systems. In specific, the study contributes to the field of systems development by the following:

1. Presenting an overall approach to systems development based on the integration of a modified, extended version of RUP with an extended version of PD.
2. Suggesting the *MOPT-Systems Development Process* based on the presented approach. The notion of MOPT systems refers to systems that consider method, organisation personnel and technology in parallel during the development process. (Figures 1–2)

More concretely, the MOPT-Systems Development Process is based on a subset of RUP principles, artefacts and notations, in combination with principles of active user participation and social research methods. It is intended to be applied in the development of socio-technical systems, with a particular focus on systems that are large, complex and distributed.

2 Background

This section presents the systems view and development approaches of relevance for the study, including socio-technical systems, the Rational Unified Process, and participatory design. Further, a brief description of the study context is provided.

2.1 Socio-technical systems

The socio-technical system view emerged in the 1970s as an opponent to the more down-right technical perspectives that, thus far, had dominated systems development thinking. According to the socio-technical view, systems consist of individuals, social, cultural and organisational components, in addition to mere technology. For systems to work well, they must fit closely with organisa-

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

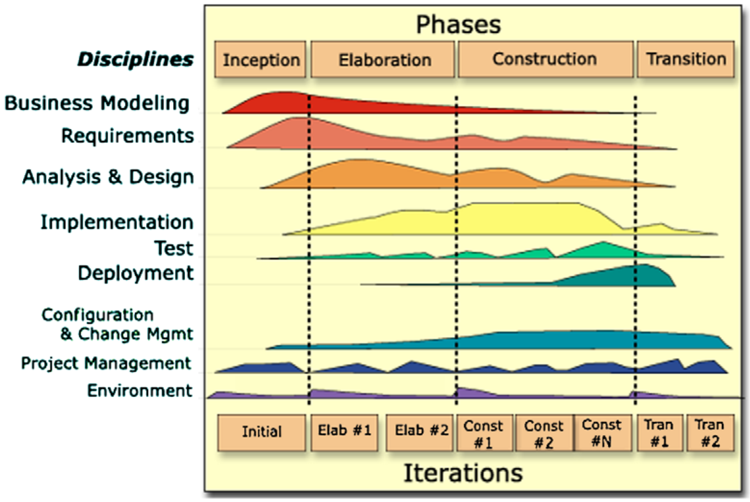

Figure 1 Example of relations between disciples and phases as described in RUP Version 2003.06.00. The different disciples receive different amount of attention depending on the current project phase

tional and social factors, and preferably enhance the quality of work life for the users. Examples of the socio-technical system view are the Effective Technical and Human Implementation of Computer based Systems (ETHICS) methodology, the Soft Systems Methodology (SSM) (Avison and Fitzgerald, 1995), and PD. The MOPT-Systems Development Process adopts a socio-technical system view which is most visible in that method, organisation, personnel and technology are seen as equally important parts of the system and are developed in parallel throughout the development process.

Of course a socio-technical perspective can be adopted for developing small as well as large information systems. In contemporary society an ever increasing amount of existing systems is large-scale and distributed, affecting many people and institutions, as well as being complex, involving many administrative, organisational, legal, political and ideological issues to be considered in the systems development process (Oostveen and van den Besselar, 2004). The proposed integrative approach follows in this development, which means that the process in specific targets (but is not limited to) development of socio-technical information systems of reasonably comprehensive size and involving heterogeneous user groups. This is, for instance, reflected in the extended version of PD and the use of social research methods.

2.2 *The Rational Unified Process*

The Rational Unified Process (RUP) is a software engineering process aimed at guiding software development organisations in their endeavours to create solid software. According to RUP, a system's lifetime is described as a finite number of development cycles. Each development cycle is divided into the four project phases *Inception*, *Elaboration*, *Construction* and *Transition* (0). The phases, in its turn, are divided into a number of iterations, depending on the project's needs and size. RUP includes nine disciples that are iteratively

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

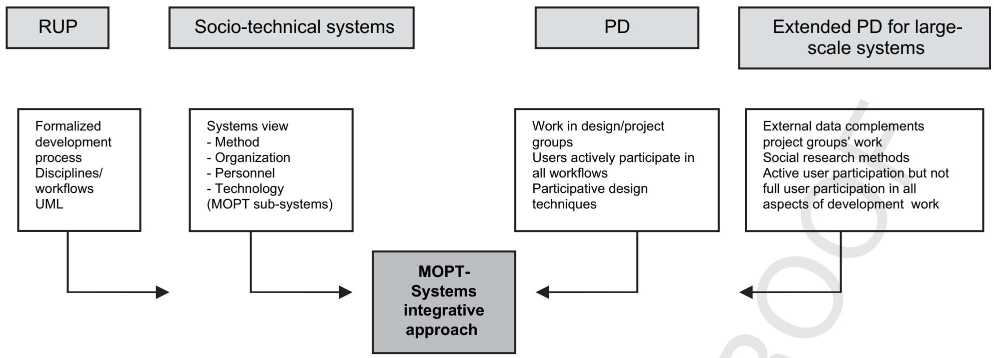

Figure 2 The different influences on and specific contributions to the MOPT-Systems Integrative Approach

executed during the different phases. The disciples are divided into technical and supportive disciples. The technical disciplines include *Business modeling*, *Requirements*, *Analysis and Design*, *Implementation*, *Test* and *Deployment*. The supportive disciplines include *Project Management*, *Configuration and Change Management* and *Environment*. Together the latter provide the infrastructure that every project needs for the project work to proceed smoothly (Kruchten, 2004).

RUP has, in the last decade, been increasingly recognised, as an efficient process for developing software intensive systems. However, while RUP suggests a process for developing the software system, aspects of surrounding organisational issues and active user participation are less well covered (Hesse, 2003; Olsson, 2004). RUP should instead be viewed as a process framework or even "a collection of good advice" that, indeed, acknowledges the potential development of the enterprise and has, in recent versions included business use cases as complementing the original system use cases (Kruchten, 2004), and which also claims to involve system stakeholders in parts of the development process. But all these aspects receive limited attention and are only loosely connected to the overall RUP process framework, as one possible way of doing things, rather than being fully integrated with concrete development work. In RUP, a use case is a complete sequence of actions related to the task the user aims to perform, either directly related to the system or to a certain work task. Use cases can subsequently be integrated into use-case models. As development proceeds, use-case models are transformed to system models, object models and so on (Kruchten, 2004). The business use cases are, thus, supposed to be an aid when initially modeling the present enterprise as the setting for the system. Change aspects, where they occur, mainly relate to officially acknowledged changes than to in-depth investigations of true end-user needs. RUP thereby, in reality presumes a rather narrowed and manager oriented focus on the organisation (Hull et al., 2001). This is much in contrast with the socio-technical system view and with PD approaches which explicitly

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

embrace change of work routines based on user needs. Further, in RUP user representation is highly selective and users used as consultants, called in as information sources when needed (e.g. as enterprise analysts) but with no real influence over the entire development process (Constantine and Lockwood, 1999). This is in sharp contrast with the continuous, active and substantial involvement of the users as recommended of the socio-technical design approaches including PD (Gulliksen et al., 2003).

2.3 Participatory design

The participatory design (PD) approaches originated in the 1970s and 1980s, when they were used as a means to empower workers at the workplace by letting them take part in the design of the technology they were going to use. The intention was to enhance workplace democracy and realise the 'good work' objective, i.e., to increase worker autonomy, skill and task variety (Ehn, 1993). PD is, thus, a rather loosely connected set of philosophies, principles and techniques belonging, to the socio-technical system view. The development of organisations and personnel are seen as equally important to development of technology and users are to be given direct influence on all aspects of the systems development process, through their participation in design groups.

PD uses a range of techniques that are supposed to be easy-to-learn and put low demand on the users' beforehand knowledge. Commonly used are mock-ups, future workshops, prototyping and scenarios (Ehn et al., 1996). Mock-ups are paper based models of the current organisation or systems. Future workshops involve the users in the systems development process, by letting them (1) reflect on their own work situation (*critique phase*), (2) identify futuristic and innovative solutions to experienced problems (*fantasy phase*), and (3) transform these solutions to be realistic and possible to implement (*implementation phase*) (Kensing and Halskov Madsen, 1991). Scenarios are constructed use situations with reference to the users' work tasks, often in a textual format (Carroll, 2000). They can be applied in the evaluative parts of a systems development process and for identifying needs and requirements. Prototypes, i.e., models of the system under development, are crucial for iterative systems development and PD (Avison and Fitzgerald, 1995; Ehn et al., 1996).

Since its origin, PD has been extended and applied also outside the immediate ideological context (Reich et al., 1996). It has been argued that it results in better systems than other approaches, since the systems are designed together with the users instead of merely using them as information sources (Bravo, 1993). But PD has also been criticised in several respects, e.g., for lack of formalisation, resulting in increased overall complexity of implementation, for extensively dealing with the early design phases while putting less emphasis on, the later, more technical stages (Tollmar, 2001), and for having a pro-longed focus on consensus reaching and democratic processes, thereby hampering efficiency and a coherent system architecture (Doll and Deng, 1999; Asaro, 2000). In spite of the

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

fact that PD now has been on the systems development arena for over 30 years, it still has not expanded to reach industrial development, neither to the development of large-scale, complex and distributed systems (Tollmar, 2001). An explanation for the latter is that the approach is only suited for small user groups in limited parts of organisations, and thus, to the design of small-scale systems. But neither small-scale PD projects have had any impact on industrial systems development. Several scholars have, indeed, critically investigated their own approach and found it, in certain respects to be an academic construction, difficult to put into industrial practice (Kensing, 2000). In conclusion, it is possible to say that while RUP is based on a marked technocratic system view, PD represents the opposite ideological and socio-technical perspective. Both these system views have their advantages and drawbacks; combining them is one way to bridge the gap between what is desired and what is feasible.

2.4 Study context

The development of the MOPT-Systems Development Process has taken place in the context of developing command and control systems for the Swedish Defense. In Sweden, a major reconstruction of the defence is currently taking place, involving the introduction of a networked, flexible and dynamic structure, built on temporary configurations of the armed forces suited to the particular mission. This, in its turn, has created a need for new and more flexible systems supporting command and control. The systems are seen as socio-technical, involving method (M), organisation (O), personnel (P) and technology (T). The development of the MOPT-Systems Development Process has received input from several projects developing control and command systems for the Swedish military ground forces and for the helicopter battalion. The projects spin over a time from the late 1990s to currently.

3 Methods and perspective

The perspective taken in the study is, first, to present an overall integrative approach for developing large socio-technical systems and, second, to suggest a specific development process; the MOPT-Systems Development Process, using a modified version of the Rational Unified Process together with supporting principles, methods and techniques from extended Participatory Design. In the study context, socio-technical systems are systems where methodology, organisation, personnel and technology are seen as closely interrelated and developed in parallel. The study is thus descriptive and includes a theoretical adjustment of chosen aspects in RUP and PD into the integrative approach and process.

3.1 Theoretical adjustment of RUP and PD

The theoretical adjustment of RUP and PD is based on:

1. The participating researchers' previous experience from PD and software engineering projects. Of the researchers, two have solid experience from managing PD projects and design groups, both for small-scale and

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

large-scale systems (Pilemalm, 2002). The third researcher has worked in several industrial software engineering projects.

2. Literature studies of RUP and PD.
3. The collected experience from previous and current projects developing command and control systems for the Swedish military ground forces and helicopter battalion. In the ground forces projects, active user involvement took place in the context of developing a system for geographically distributed command and control. In the helicopter battalion project RUP and PD principles were explicitly combined in developing a command and control system for the entire Swedish helicopter battalion. Two of the researchers in the study, one with PD experience and the one with industrial software engineering experience worked in some of the ground forces projects and also accessed documentation from all these projects in retrospect. As for the helicopter battalion project they both acted as systems developers in the project design group and initially applied the MOPT-Systems Development Process. In this way, they were able to gather their experience and continuously feed it back to the process.

3.2 Documentation

The overall MOPT-Systems Development Process was constructed based on the experience as described above, and documented in a report. As the MOPT-Systems framework had been initiated before the helicopter battalion project, the impressions of the two researchers participating in this project were discussed after each project meeting, to create a common picture, and used as input to modifying the process. As for this study, this regards, above all, enterprise modeling, needs analysis and requirements engineering (the project is still in progress). But also the process in its entirety has been continuously updated to reflect new findings, thoughts and ideas.

4 Results

In this section, the results of the study are presented. First, the overall approach is described on a theoretical level, displaying its major cornerstones collected from RUP, the socio-technical systems view and PD and comparing it to the others. Second, a more detailed description of the proposed MOPT-Systems Development Process is provided including descriptions of the process workflows, the process in practice and some toolbox work tools.

4.1 Description of the overall MOPT-Systems approach

The MOPT-Systems approach is based, first, on modifications of chosen aspects of RUP. This includes, above all, iteration in and between nine workflows. Further, the RUP Unified Modeling Language (UML) which has a defined syntax, including, e.g., objects, relations and diagrams and further includes mechanisms for extending syntax and notations (Cockburn, 2001) is used to support the use case based modeling parts in the process

(Oesterreich, 2002; Kruchten, 2004). In comparison to RUP, activities in the workflows have, however, been extended in the sense that method, organisation, personnel, and technology are developed in parallel throughout the development process; in order to reflect the socio-technical perspective. Some workflows relate mostly to the system context, some mostly to the overall system, and some to its MOPT sub-systems. The iteration and traceability between workflows is crucial, in order to facilitate verification, validation and further development of the ensuing system. Further, the approach integrates a subset of existing principles and techniques from PD. The approach taken here is an extended version of PD where traditional techniques for user participation such as, e.g., future workshops, scenarios and prototyping, have been complemented with social research methods in order to reach a wider user group. This is in line with recent attempts to develop PD to suit development of large-scale systems and large, heterogeneous user groups (Pilemalm, 2002; Oostveen and van den Besselar, 2004). Examples of social research methods applied include, for instance, Critical incident technique questionnaires (CIT) and the Critical decision method (CDM). CIT was originally an interview technique used to recruit American Air Force pilots by investigating how they would behave in critical situations (Flanagan, 1954). It has since been used in a number of fields, including systems development (Dean, 1998). Here, CIT has been shown effective to capture breakdowns in work activities and in the next step to identify solutions and remedies to the problems, i.e., to identify user needs and system requirements. The CDM is an off-spring from CIT and is a specific interview technique that aims to identify the process for decision-making in critical, non-routine incidents taking place in dynamic environments characterised by time pressure, high information content and changing conditions (Klein et al., 1989). illustrates the different inputs to the overall MOPT-Systems approach from the other approaches.

Similarly, Table 1 provides a comparison of the overall approach as related to RUP and PD. It also functions as a summary of the ensuing suggested MOPT-Systems process.

4.2 The MOPT-Systems Development Process and its workflows

Based on the overall approach, a more formalised process is suggested and substantiated as *the MOPT-Systems Development Process*. The process is defined by nine workflows, to a certain extent reflecting the nine disciples in RUP. However, while RUP divides the disciples into technical and supportive disciples, all the workflows in this case treat the Systems Development Process. The supportive infrastructures that surround every systems development project are treated separately as part of the practical application of the process. The nine workflows of the MOPT-Systems Development Process are, thus, *contextual modeling, needs analysis, system requirements engineering, system analysis and design, sub-system development and implementation, system*

Table 1 The MOPT-Systems approach, as compared to the Rational Unified Process and participatory design

	Rational Unified Process	Participatory design	MOPT-Systems approach and process
Scope of systems development process	Covers entire development process for software engineering.	Covers only early phases in development process for socio-technical systems.	Covers entire development process for socio-technical systems.
System target group	Large software systems in an enterprise.	Small-scale socio-technical systems in an organization.	Large socio-technical systems in a wide system context also embracing surrounding environment.
Systems development process	Iteration in and between nine disciplines in each project phase. Division of technical and supportive disciplines.	No defined disciplines or workflows. Process is loose and context dependant.	Iteration in and between nine systems development workflows in each project phase. Supportive disciplines not included but part of the MOPT-Systems practical application.
Focus of systems development process	Software system in focus. Subsequent focus on the system requirements, technology and architecture. No explicit needs analysis.	Socio-technical; end-user and their real needs, system, and organizational context in focus.	Socio-technical; the enterprise/context and the system to be developed simultaneously. Models context in which the system will function. Develops methodological, organizational and personnel sub-systems simultaneously to technological sub-system (MOPT). Performs needs analysis.
Enterprise/organization/ context view	Relatively unproblematic and manager oriented view on enterprise. Do not explicitly emphasize needs for organizational change.	Conflict view. Emphasizes the identification and resolving of existing conflicts and problems in organization, especially between managers and workers/users.	Emphasizes needs for change aspects; existing problems and future changes in mind when modeling system context.
User participation	Implicit assumption that system stakeholders can participate in parts of systems development process; but no explicit guidelines as how to achieve active involvement of users in entire process.	End-users participate in all aspects of design work, including planning and decision-making about work routines. Development takes place in design groups.	End-users work actively in project group throughout entire development process. A trade-off of work between users and systems developers where users provide information on context, user and system functionality needs while systems developers are responsible for, e.g., project planning, decision-making and documentation and detailed design and implementation.

(*continued on next page*)

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

Table 1 (*continued*)

	Rational Unified Process	Participatory design	MOPT-Systems approach and process
Practical guidelines for development process	Describes the general process framework and include "how to do it" practical aspects.	Has no specified process but provides a philosophy, and a set of principles and tools.	A suggested and specified systems development process complemented with "how to do it" general guidelines.
Tools for data collection	Provides examples of ways to collect data for modeling the enterprise and identifying requirements but few explicit user-oriented tools and no tools for capturing user needs.	Toolbox existing of a range of PD techniques.	Toolbox existing of a range of PD techniques and social research methods for complementary data collection from the user groups and stakeholders who are not represented in project group.
Tools for structuring information	Applies the UML notation language for modeling.	Lacks coherent notations or commonly agreed upon modeling tools.	Applies the UML notation language for modeling.

integration, system verification, deployment and *system validation*. One or sometimes several project groups consisting of systems developers and user representatives work together through all the workflows. The UML notation is used for all modeling.

4.2.1 Contextual modeling

When developing socio-technical systems, understanding the context and environment in with the system will be operative is crucial. In the MOPT-Systems Development Process, contextual analysis takes place by contextual modeling. The overall activity performed in this workflow is, thus, to develop a model describing the organisational context in which the system will serve. This is achieved by collecting data and using it as input when modeling the context activities as use cases. The active role of the users is crucial in this activity as they possess the domain knowledge. Information should be extracted both from the user representatives in the project group and from users, domain experts and stakeholders external to the group. When modeling the context the perspective is on the present as well as on the future. This implies that also current difficulties, problems and needs for change should be explicitly captured, in-depth analyzed and incorporated into the modeling work. Here, methods and techniques for capturing change aspects, as for instance Future workshops and the Critical incident technique can be used.

Workflow input: Data about current organisation, its immediate context and surrounding environment.
Workflow output: Contextual model reflecting the future organisational context and environment that the system should support.

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

Degree of active user participation: High.

4.2.2 Needs analysis

While RUP proceeds directly from Business Modeling to Requirements and only briefly treats needs as *stakeholder requests* (Kruchten, 2004), the MOPT-Systems Development Process, introduces the intermediate workflow of *needs analysis*. This is done in order to capture the context's and users' true needs of the system, and to avoid a too early focus on solutions and fixed requirements. Many of the needs can be identified directly by the system developers using the contextual model. Further, reflecting over the model in a project group can help the users participating in the group to identify new needs. In the latter case, the contextual model becomes an initial prototype wherefrom needs are identified, distilled, refined and concretised. Additional needs are captured using social research methods and PD techniques (see Section 4.5). The established needs are further analyzed, structured, prioritised and documented in a database with traceability to their sources and preceding statements (see Section 4.5).

Workflow input: Contextual model, social research methods and PD techniques.
Workflow output: A document specifying the system context and its users' needs and their priorities; a needs description.
Degree of user participation: High.

4.2.3 System requirements engineering

The workflow handles requirements on the ensuing system. The overall activity performed is much similar to the requirements handling in RUP, but with a great amount of requirements extracted directly from the needs specification. Additional requirements are collected in order to complement and fill in missing requirements or those requirements that are not explicitly preceded by a need. The identified requirements are analyzed, structured, prioritised and documented in a database with traceability to needs or statements. The project group and other user representatives play a continuous important role in identifying and prioritising requirements. A low-fi prototype should be developed at this stage and used to support the users in evaluating the requirements. In the requirements documentation, categories and sub-categories of requirements are displayed hierarchically in order to facilitate overview and, to see if some category is missing or weakly represented, i.e., whether iteration is needed.

Workflow input: Needs specification, organisational governing documents, IT-security and usability expertise.

Workflow output: A model specifying the external view of the ensuing system displayed as system use cases, a requirements specification and a prototype.
Degree of user participation: High

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

4.2.4 System analysis and design

The workflow handles overarching design of the system. The overall activity performed is to determine the system architecture, design principles, and decisions. A core activity in this workflow is to determining which requirements are of interest for the sub-systems development, i.e., M, O, P, T. The workflow also re-uses the system use cases to identify a set of objects. The objects are subsequently defined into classes, using UML class diagrams. The project group and user representatives are somewhat less involved in this workflow, since much of the work requires modeling, systems architecture and design skills. Still end-users are, as compared to RUP, much more involved in the design process, mainly by means of iterative prototyping. Prototypes at this stage are often broad with rather low functionality, focusing on and testing different design and architectural solutions.

Workflow input: Requirements specification.
Workflow output: A model describing the internal view of the system and its objects and classes, prototypes and an overall design description.
Degree of user participation: Medium.

4.2.5 Sub-systems development and implementation

Because of the socio-technical perspective, a crucial aspect of the MOPT-Systems Development Process is the sub-system development. While RUP has its overall focus on the software system, the current process continues to consider method, organisation and personnel in parallel to technology. The overall activity performed in this workflow is thus to develop and implement the MOPT sub-systems as part of the overall system. *Data collection*, *analysis*, *design*, *implementation* and *verification* are subsequently performed for *each* sub-system. The concrete content of these steps look different depending on what sub-system is being developed. For instance, implementation of technology implies coding and testing while in the case of method it means to structure and document new methods in a textual format. Also, even though the Figure 3 may give the impression of sequential development performed by different individuals; in reality activities are executed in parallel, jumps between activities take place, input from one sub-system may provide input to another, several individuals partake in the development of several sub-systems, several project groups may be formed and so on. A certain degree of parallelism is even a must, in order to achieve reasonable sub-system coherency. The project group and user representatives play an important part in the M, O, and P sub-systems while development of pure technology is more of an issue for system developers and technicians. The outcome of the activity is MOPT sub-systems with specific deliverables for each sub-system:

Workflow input: Data from previous work (e.g. needs specification, requirements specification), complementary data from users.
Workflow output: A number of deliverables for each sub-system.

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

Method deliverables: Sub-system use-case model, analysis model, requirements specification, document providing structured work methods and how they are to be deployed in organisation and integrated with system. Activity diagrams.
Organisation deliverables: Sub-system use-case model, analysis model, requirements specification, document providing future organisational structure and how it is to be deployed in organisation and integrated with system. Class diagrams.
Personnel deliverables: Sub-system use-case model, analysis model, requirements specification, competence plans, training directives and material. Class diagrams, roles connected to position, description of competences.
Technology deliverables: Sub-system use-case model, analysis model, requirements specification, design description, software, and test reports. Realisation diagrams.
User participation: High.

4.2.6 Systems integration

The workflow deals with the integration of the MOPT sub-systems. As compared to RUP, where integration is seen as part of the implementation workflow and where system components from individual implementers or teams are integrated, integration here has its own workflow. The overall activities performed are to successively and incrementally integrate the MOPT sub-systems to an entire, functioning socio-technical system, to plan and perform tests. This implies, in practice to solve those problems and conflicts that have not already been solved during sub-system development. For instance, technology may be modified and configured to support new work routines or, the other way round, some methods may be abandoned since they do not fit in with fundamental technological requirements. Prototypes of the system are developed and successively added with more functionality and components before deciding on the final system solution, and hence, users play an important part in evaluating different versions of the system. It should be noted that for large-scale systems (as emphasised also in RUP), it is most often better to perform development incrementally, i.e., to develop several sub-systems and components that are successively added and integrated, rather than to aim at a total solution from the very beginning. The project group and user representatives are also needed when prioritisation of contradicting sub-system issues must take place.

Workflow input: The MOPT sub-system and possible other sub-systems and components.
Workflow output: Socio-technical system and test models results.
Degree of user participation: Medium.

4.2.7 System verification

The workflow aims at verifying the socio-technical system before it is deployed into the enterprise. While RUP uses the test discipline for covering both verification and validation the MOPT-Systems Development Process

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

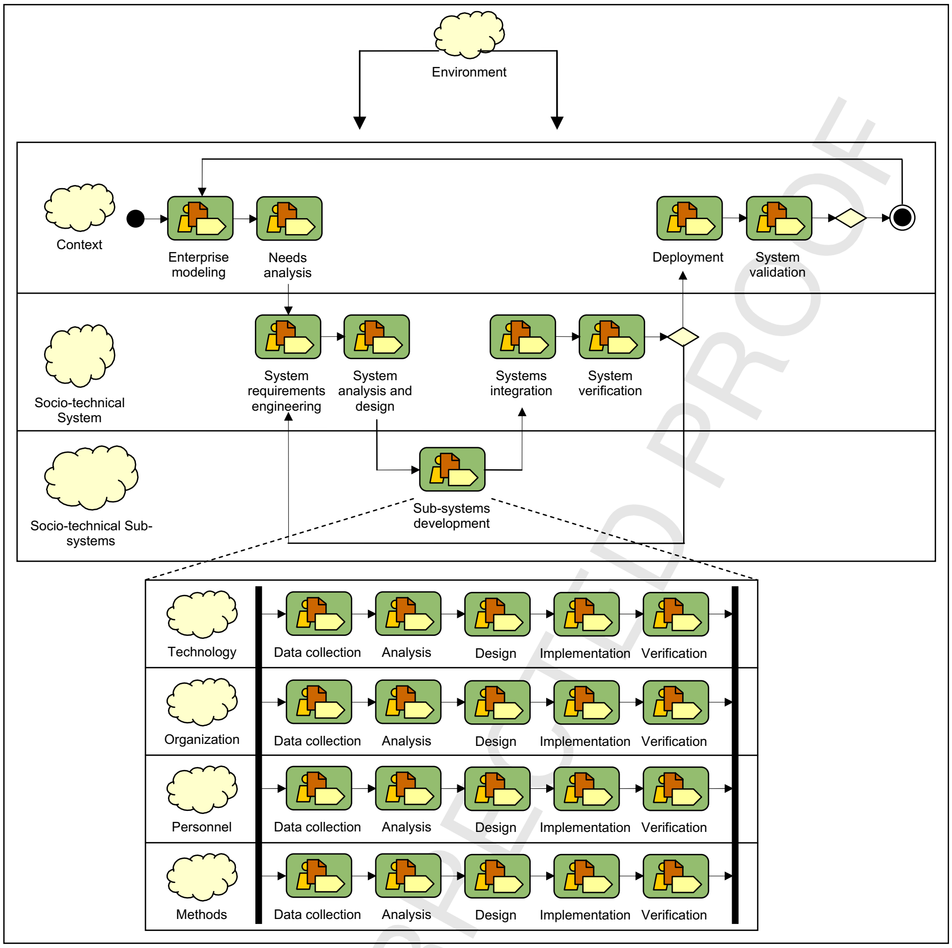

Figure 3 The nine workflows of the MOPT-Systems Development Process. Iteration takes place as needed in the current project context. Possible iterations are exemplified with a point of departure in plausible decision points

distinguishes between the two in order to reflect the previously made distinction between needs and requirements. The overall activity performed in the verification workflow is to verify that the system fulfils its requirements considering technology as well as organisation, method and personnel. More concretely, the requirements substantiated in the system are compared with those (prioritised) requirements in the requirements specifications. If there are contradictions, it must be decided whether this is a result of acceptable modifications made during system integration or if the system must be extended or modified. End-users play a less prominent part in the verification workflow but the project group should be provided with feedback. A verification model and report provides the basis for a decision point, i.e., a decision whether the system is correctly built (fulfils the system requirements) or if more iteration is needed.

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

Workflow input: Requirements specifications both for entire system and for the MOPT sub-systems.
Workflow output: Socio-technical system, a system verification model and report.
Degree of user participation: Low.

4.2.8 Deployment

The aim of the workflow is to deploy a functioning socio-technical system into the enterprise. The major activity performed is thus to install the system and, if necessary, carry out modifications to make it work. All the MOPT sub-systems have to be deployed; deployment thus implying different things for each sub-system. As for technology, software and hardware are installed, or software integrated with existing hardware. Further, additional tests are performed testing the system against pre-defined criteria concerning, e.g., speed, capability and robustness. As for methodology and organisation, new methods and work routines are anchored in the enterprise, through the personnel responsible to carry out required changes. As for personnel, the end-users are trained in the system and in other needed competences as identified in the current sub-system. This workflow, thus, has the entire "real-life" enterprise end-user group in focus.

Workflow input: Socio-technical system.
Workflow output: Deployment reports including problems, solutions and experience, and the enterprise managed by the new socio-technical system.
Degree of user participation: High.

4.2.9 System validation

The MOPT-Systems Development Process includes system validation as the last workflow. The activity performed is to validate if the system actually fulfils the needs of the users. The major input to workflow comes from the needs analysis workflow as well as from the actual experience from the system in practice, i.e., the experience of the end-users. The latter is captured by interviews, observations of how end-users use, interact with and acknowledge the system, and by simply talking to the users. The "real-life" enterprise end-user group is thus in focus. During the workflow, new needs may be identified or those previously identified modified to update the system or to be used in a new release of the system. A validation report provides the basis for a decision point, i.e., a decision of whether the system fulfils the needs of the enterprise and its users or if more iteration is needed.

Workflow input: Needs specification and experience of system in real use.
Workflow output: Socio-technical system and a validation report.
Degree of user participation: High.

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

Figure 3 visualises the MOPT-Systems Development Process. In the process, context is the immediate system context, most often denoted as organisational context. System is the socio-technical system, in itself embracing those parts of the organisation that are directly affected by the system, with related methods and personnel aspects. Environment is external to the immediate context but may still have an influence on the system through different actors and interests. Further, it should be noted that even though the workflows are described in sequence and may denote a top–down hierarchical process, this is not the case in reality. Iteration is a crucial aspect of the process and several workflows are supposed to be executed in parallel and switched between, providing input to each other and partly involving the same individuals. As an example, new needs and requirements may be established during the evaluation of prototypes, implying that not only the needs analysis and requirements engineering workflows needs to be iterated, but also that the contextual model has to be updated, reflecting new work routines. In line with iteration and as compared to RUP, no explicit project phases are described in the process. This decision is deliberate, reflecting recent critique on RUP as in reality copying waterfall-like models and having a monolithic view on the development process (Hesse, 2003). In the current process, the workflows instead evolve around management of the different decomposed units and sub-systems that are present in large, complex systems.

4.3 The MOPT-Systems Development Process in practice

In the above descriptions, the roles and participation of the users were briefly described in relation to each workflow. The MOPT-Systems Development Process further suggests a practically oriented framework that provides a number of general, concrete guidelines for how the process can be applied in real systems development projects. The concretisation elaborates on project organisation, including the aspects of how active user participation and the added extended PD aspects work in practice. The framework further includes a toolbox of PD techniques and social research methods to support concrete development work.

4.4 Project organisation

According to RUP, teams of individuals work together during the software engineering process. In the teams individuals acquire certain roles such as, e.g., system analysts, business designers and requirements engineers. The association between a role and an individual is not fixed; several individuals can acquire the same role and vice versa. Neither are the teams as such stable as different roles are executed during different phases of the development process (Kruchten, 2004). The role of the user is not explicitly handled in RUP even if it is possible that, for instance, one or several of the context analysts are user representatives from the current enterprise acting as domain experts. PD, on the other hand, assumes that user representatives actively partake in the entire development process and even dominate a design group. In the current

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

process, a project group is formed, consisting of system developers and user representatives. Several of the roles in RUP are re-used and responsibilities divided between them. But the process also adds the explicit role of user representative, where a project group should involve of at least two to three user representatives for each system developer. The established group then works together through the entire MOPT-Systems Development Process (in an ideal case, pre-cautions to handle eventual participant turnovers must always be taken; Pilemalm, 2002).

However, designing large-scale systems using a PD perspective involves certain difficulties as to reach the entire user group. It has also been demonstrated that full user participation in all aspects of the development process neither is efficient nor in agreement with user satisfaction (Doll and Deng, 1999). In recent attempts to extend the PD approach, parts of the development work has therefore taken place outside the immediate design group with data collection from different user groups applying social research methods (Pilemalm, 2002; Oostveen and van den Besselar, 2004). In the MOPT-Systems Development Process, there is a division of work between development taking place in and external to the project group. The project group is kept permanent to the extent possible. A number of systems developers (depending on project context) manage and coordinate the work of the group throughout the development process. These persons can, for instance, have the role of context analysts or system analysts.

Work of the user dominated project group includes, e.g., analyzing the context, identifying system stakeholders, identifying and prioritising needs and system requirements, performing design practices and evaluation of prototypes, and providing general feedback on all results produced in the systems development process. The work where the project group user representatives are less involved includes data collection from identified important additional user groups and stakeholders using various social research methods. However, the gained results are always fed back to the entire project group. The data collected external to the group have two functions; first to verify the findings of the group against a larger data material (e.g. by questionnaires sent to many respondents); and second, to collect the voices of those users and stakeholders that are not directly represented in the project group. Through this combination, entire organisations can partake in designing large-scale systems while the project group is always present providing feedback. Other activities in the development process which are less of candidates for user participation include e.g., administrative and documentation work in project group (participating system developers responsible), detailed modeling (analysts responsible), detailed needs and requirements engineering including management of requirements in a database (requirements engineers responsible), detailed design (designers responsible) building system architecture (system architects responsible), integration (implementers responsible) and so on. This implies

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

that, in above all the later phases of the development process, individuals possessing these roles must, besides from performing this work outside the project group, also act as guest players in the group presenting their results (for instance a prototype), and receive the necessary feedback from the user representatives. In this way, the project group can be kept reasonable stable while interaction between internal and external development work remains. The users still have the opportunity to influence important issues in the development process, even though to a different extent in different workflows. Also, in large project contexts, it may be necessary to form several project groups, from the very beginning or as the process grows more complex (but at least one group needs to follow the entire development process), to put a reasonable load of work on the group members and to expand on the direct involvement of end-users. This, of course, involves additional tasks for the systems developers as to coordinate and integrate the results of the different groups. In conclusion, the MOPT-Systems process thus adheres to the principle of active user participation throughout the workflows, but not necessarily to full user participation in ALL aspects of this work.

4.5 The toolbox

The toolbox consists of RUP tools, PD techniques and social research methods. The reason for including social research methods is that the PD project group, when designing large-scale systems, is too small to adequately represent all user groups and stakeholders. Likewise, the RUP notion of enterprise has a wider connotation here and is to be seen as the entire system context. In the following section, the usage of some toolbox tools, techniques and research methods is exemplified and motivations of why they where chosen provided. This, of course, does not elude other methods or techniques. The specific combination is dependent upon the prerequisites for each project.

4.5.1 RUP tools

The RUP tools included in the toolbox are the *The Unified Modeling Language* notation and the *Rational Rose* support tool.

The Unified Modeling Language (UML): In the MOPT-Systems Development Process UML is used in modeling use cases and models. As for the latter, the process applies context models, system models, sub-system use-case models, test models and verification models. The UML was chosen in order to provide notations that support structure of data and process, in order to arrive at a formalised development process.

Rational Rose: Rational Rose is a commercial software development support tool aimed to sustain RUP (*RUP Version 2003.06.00.*). It supports the UML modeling and further provides, for instance, automatic document generation and a database (*Requisit Pro*) where statements, needs and requirements can be stored and traced to each other. The MOPT-Systems Development Process

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

applies Rational Rose in the above respects, to bring structure and overview to the process.

4.5.2 Social research methods

The social research methods included in the toolbox are a number of traditional qualitative data collection methods, for instance questionnaires and interviews.

In the field of systems development, *questionnaires* can be used to acquire information on users' work situation and tasks and thereby pinpointing their needs and system requirements. In the MOPT-Systems Development Process different types of questionnaires are applied for different purposes and sometimes in combination with prototyping. In the main, the questionnaires have open-ended questions (Taylor and Bogdan, 1998) seeking out users current work situations, problems, needs, and experiences from practically working with the system under development. One example is the *Critical incident technique (CIT)* which is applied in the project group to capture the users' experienced problems in the context and work situation, and to work out solutions to these problems. Solutions may be of an M, O, P or T nature. Further, CIT questionnaires are sent to additional user representatives and stakeholders external to the project group. CIT has in previous projects shown to be extremely useful for providing rich descriptions of users' present situation, needs and requirements, and can also be used to reach a large number of respondents from where general patterns can be extracted.

In systems development *interviews* are suitable for reaching the perspective of important individual actors and stakeholders in a system. In the MOPT-Systems Development Process, interviews are applied for collecting organisational knowledge, needs and requirements from users and stakeholders not represented in the project group. For instance, it might be necessary to interview those people who are to administrate or finance the system. Interviews are also used for evaluation of the system, often in combination with prototypes. They are, most often, of a semi-structure character where theme questions are formed to guide the seeking for needs and requirements, as well as for identifying constraints on and subjective experiences of the system (Taylor and Bogdan, 1998). In specific, *The Critical decision method (CDM)* is applied when decision-making and command aspects are crucial. The method has shown effective when the system is to provide decision-support for non-routine incidents (Klein et al., 1989).

4.5.3 Participatory design techniques

Examples of explicit design techniques with a high degree of user participation include Future workshops, prototyping and scenarios.

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

In the MOPT-Systems Development Process, *Future workshops (FW)* are applied in the project group, sometimes in combination with the previously established contextual model (which then serves as a trigger for the critique phase). Future workshops are also held with other user groups, both to complement and corroborate the findings of the project group. Future workshops have in previous projects been easy for the users to comprehend and successful in starting out with their everyday reality and successively forming needs and requirements.

RUP only briefly describes *prototyping* as a way of reducing risks and uncertainties regarding the system. It does not clearly integrate their belonging in the development process (Kruchten, 2004). The MOPT-Systems Development Process, on the other hand, makes extensive use of prototyping and includes the construction of prototypes from the needs analysis and system requirement workflows and forward on. Prototypes are evaluated both in the project group and by other users and stakeholders, with the aims of finding needs, requirements and identifying suitable design solutions. Prototypes are included to complement the abstract UML models with concrete hands-on experience. Previous project experience has shown that the possibility to work with concrete tools, yields not only rich information results but also enhances user satisfaction in a group.

In the MOPT-Systems Development Process, *scenarios* are constructed using the context model as input. The scenarios are used for condensing and concretising the model and thereby triggering the identification of needs and requirements in the project group. They are also used together with prototypes and for evaluation and validation purposes. The motivation for including scenarios in the toolbox is that they are useful triggers for needs and requirements and further support working with the somewhat abstract context model in a more concretised manner, more graspable for the users.

5 *Discussion*

The approach presented in this study was developed in response to a perceived need to combine the benefits of a socio-technical system perspective using an extended version of PD together with a more structured systems development process based on a subset of RUP principles and covering also technical aspects. The major modifications of RUP included, for example, a change of name for some terms, introducing the MOPT sub-systems and letting the nine workflows all relate to the systems development process rather than to treat some as supportive disciplines. The latter was felt necessary in order to elaborate and add to the development process, above all when introducing the needs analysis and system validation workflows. This does not imply that the supportive aspects are neglected but rather that they belong to the practical guidelines for applying the MOPT-Systems Development Process. As for the introduced needs analysis and corresponding validation against

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

needs, it was felt important that these workflows receive emphasis in the process, in order to capture users' real everyday problems and needs. The difference between a need and a requirement is sometimes subtle but in other cases substantial. Focusing on requirements at too early a stage leads to a risk of thinking in terms of existing technological and organisational constraints as well as suggesting advanced technological solutions, rather than grasping what is really needed. Further, needs are often more easily acknowledged by the users than are requirements, since they relate to their every day context, rather than to an abstract system. The introduction of a needs analysis thus improves the prospects for active user participation. As for the MOPT subsystems, introducing them is the largest modification to RUP and an extension deemed as absolutely necessary to arrive at socio-technical systems usable and well-functioning in their context.

In the study, it is not argued that RUP completely eludes aspects of involving the end-users and the organisation surrounding the system. However, RUP (RUP, *Version 2003.00.06*) provides merely a general framework in which these aspects are acknowledged as possible, but are not emphasised. In other words, the approach tells what can be done rather than how to do it. Explicit participatory design techniques are not mentioned. Also, later RUP versions briefly mention change aspects when modelling the enterprise but this seems to relate to officially acknowledged changes put forward by management rather than to true change needs. PD, on the other hand, provides concrete tools that guide active user participation in a development process aimed at changing and improving organisation and work routines as well as technology. On the other hand, also PD has it drawbacks when it comes to structure, formalisation, technology and end-product. In conclusion, the RUP inspired formalisation of the MOPT-Systems Development Process, leads to that the risk of focusing to extensively on early development (a risk that is indeed substantial in "non-pd" projects as well) is avoided. The process adheres to the principle that the work focusing on technology (implementation, integration, deployment, etc) should receive about two thirds of the total time and resources in a systems development project. The remaining time, if used efficiently and in a structured and planned manner, should suffice to include a solid context analysis, performing needs analysis and requirements engineering.

5.1 Practical experience

So far, the three first workflows of the MOPT-Systems Development Process, *context analysis*, *needs analysis*, and *system requirements engineering* have been practically applied. This in the helicopter battalion project aimed at developing a command and control system for the entire Swedish helicopter battalion. Work has taken place in a project group, consisting of systems developers and user representatives from command personnel to helicopter crew. The users, from the outset, have played an active role in the development process; by

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

(1) participating in the modeling of the context and system, i.e., the helicopter battalion with a special focus on command and control issues, (2) identifying the needs of the battalion, with a point of departure of the emerging contextual model, and by performing the critical incident technique, Future workshops, and scenarios for capturing needs and requirements, and (3) identifying corresponding and additional requirements on the system. The experience is that the major bulk of the work when describing the context could be performed within project group meetings, much of the modeling being performed in real time. The group made use of the user participants' knowledge and of documents describing, above all, the command and control process. However, some activities where not covered by the experience of the group, neither by the literature. An example is international missions, an activity which is deemed to become crucial in the future Swedish defence. Here, the systems developers had to seek additional data by performing both interviews with experts and by letting experts participate in the modeling of international missions at a separate meeting. Also, the user representatives in the project group, at the outset, deemed the context model as somewhat abstract and incomprehensible. This was solved by adding comprised textual explanations to all models. The produced context model reflected the helicopter battalion as it will basically look like in ten years. In this specific case modeling the future was not considered a difficulty, as quite clear guidelines for how the battalion will develop (for instance trough the purchase of attack helicopters) exist. However, additional new directives for the battalion have arrived recently, requiring an iteration of the context modeling workflow.

Performing the needs analysis included more of a trade-off between systems developers and user representatives. Needs were identified partly by the systems developers directly from the context model, and partly at project meetings. In the latter case, the most relevant use cases in the model were gone through in detail, and related to what needs occur when performing a certain activity or task. Further, additional needs were identified in the group by carrying out the mentioned PD techniques. An experience made is that the context model is useful for capturing user needs, but that the needs that are identified from the model needs to be further penetrated, detailed, explained and concretised, in order to move the systems development process forwards. Otherwise, there is a risk that the needs analysis simply copies what has already been found during the context modeling. The needs analysis workflow is currently iterated, due to the new directives.

As for performing requirements engineering, the experience is that initial work was somewhat slow and cumbersome as the user participants had to "learn" the fundamentals; the basic "thinking" and concepts of requirements engineering. This took longer time than expected, partly because a few user representatives had left and others entered the group as newcomers (the problem with user-turnover has much to with the present status of the Swedish defence receiving less financial support from the state resulting in cut-downs, layoffs and replacements). However, it is, in retrospect, believed that this initial delay was worth the effort, since the following activities of identifying and filtering the "true" requirements, for instance from the multitude of densely populated governing requirements documents produced by the Swedish defence, went much smoother and would not have been possible with out the domain knowledge of the users.

In all workflows, there was common agreement among the users that they did not have the time to involve themselves immensely in administration, documentation or other "homeworks". In other words, full user participation was regarded as dysfunctional and they wanted to focus on the domain knowledge they could provide and on needs and requirements generation.

It might be argued that the merging of two systems developments approaches is demanding in time and in resources. Practical experience from applying the merge has, thus far, worked smoothly with having a rather small project group and by having two systems developers perform the work and data collection between and external to project meetings. A few additional system engineers have now started to work with the project group as for the subsequent work. It is likely that more systems development resources are required in later phases, maybe even requiring additional project groups. On the other hand, building large technical systems must be ensued to take time and resources and the PD extensions of RUP will hopefully, in the end, lead to that user and organisational needs for change are truly captured, thereby resulting in usable systems and in less costly and time taking re-buildings. In this respect, also introducing the workflow of needs analysis is supposed to contribute, as a way of identifying what users and organisations really need, in contrast to them telling what they think they need and proceeding directly to requirements.

5.2 Context, limitations and future research

The MOPT-Systems Development Process is aimed at creating an effective trade-off between users, stakeholders and systems developers, and at delivering an end-product, while not losing the needs for change in organisations, neither the general benefits of active user participation. The process embraces socio-technical systems of large-scale in specific. This is, for instance, reflected in the division of work inside and external to the project group reaching heterogeneous user groups and different organisational layers in the data collection, and by acknowledging the possibility of forming several project groups, either from the beginning of or after some time in the development process. The process is thereby in line with contemporary trends of increasingly large, distributed, complex and technically sophisticated systems, involving a multitude of people. In specific, it contributes to adapt PD to this continuous growth. However, the process may as well be applied in projects of smaller

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

scale with more homogeneous users – such a context would probably even make the execution of the process smoother.

Finally, it should be noted that even though the MOPT-Systems Development Process in many aspect is a novel process, it is not to be seen as entirely pioneering work breaking new ground. The importance of combining socio-technical aspects and the end-users with more rigorous and complete processes for developing organisational yet technically complex and solid systems has been on the agenda for quite some time. The process described should instead be seen as a suggested overall approach and a process that is coherent and still flexible enough to be practically feasible. Thus, far it is only the three first workflows of the process that have been tested and no formal evaluation has taken place. Future research should test the MOPT-Systems Development Process in its entirety in a concrete systems development project. In the helicopter project, prototyping and scenario-based evaluations with the users will be applied in subsequent design. In some ensuing workflows, e.g., system integration and verification, users in the project group will probably play a less prominent role but will still be present through their continuous feedback on the emerging system. In other workflows, e.g., sub-systems development, system deployment and validation, users, in and outside the project group(s), again take an active stance by evaluating and proposing modifications to the system. Important issues to be considered in the further development of the process include how to balance user participation and the need for software engineering skills in the subsequent workflows, how to further develop the project management aspects, how to really achieve iteration in and between work flows, and how to coordinate the work of several project groups and the development of sub-systems in very large projects. In relation to this, the process needs to be further developed in respect with how to merge the data collection, analysis, design, implementation and verification performed for each MOPT sub-system with those performed for the overall system.

6 Conclusion

In the contemporary field of systems development there is an ever growing awareness about the need to involve users in the development process as well as about the importance of developing organisation, people, work routines and methods simultaneously as the technical system. Still, industrial development of information systems are still, to a great extent, performed by software intensive approaches focusing on the technology infrastructure and less on the end-users' true needs. This study is one attempt to bridge the gap between ideology and reality. The MOPT-Systems Development Process integrates the formalised, all workflows inclusive development process of RUP, though in a modified version; with socio-technical aspects of developing method, organisation and personnel hand in hand, together with the end-users and accompanied with PD techniques, as well as with a solid data collection

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

from the current system context. The expected output is a technically solid end-product that functions well for the organisation and is appreciated by its users. Thus, far, parts of the process have been applied when developing command and control systems for the Swedish defence; however, the process is supposed to have general applicability for developing information systems, in particular large-scale systems. Future research should test the MOPT-Systems Development Process in entire systems development projects, covering the process from initial organisational analysis to system deployment and validation.

References

Asaro, P M (2000) Transforming society by transforming technology: the science and politics of participatory design *Accounting, Management and Information Technology* Vol 10 pp 257–290

Avison, D E and Fitzgerald, G (1995) *Information systems development: methodologies, techniques and tools* The McGraw-Hill Companies

Bravo, E (1993) The hazards of leaving out the users in **D Schuler and A Namioka** (eds) *Participatory design*. Principles and practices Lawrence Erlbaum, Hillsdale, NJ pp 3–11

Carroll, J M (2000) *Making use Scenario-based design of human-computer interactions* Massachusetts Institute of Technology, MA

Cockburn, A (2001) *Writing effective use cases* Addison-Wesley

Constantine, L L and Lockwood, L A D (1999) *Software for use: a practical guide to the models and methods of usage-centered design* Addison-Wesley, Reading, MA

Constantine, LL and Lockwood, LAD (2002) Usage-centered engineering for web applications *IEEE Software* March/April

Dean, P J (1998) A qualitative method of assessment and analysis for changing the organizational culture *Performance Improvement* Vol 37 No 2 pp 14–23

Doll, WJ and Deng, X (1999) Should end-users participate as much as they want in the development of collaborative applications? in *Proceedings of the 32nd Hawaii International Conference on System Sciences*

Ehn, P (1993) Scandinavian design: on participation and skill in **D Schuler and A Namioka** (eds) *Participatory design*. Principles and practices Lawrence Erlbaum, Hillsdale, NJ pp 41–77

Ehn P, Davies, RC, Brattgård, B, Hägerfors, A, Nilson, J, Dalholm, E and Mitchell, B (1996) The envisionment workshop – from visions to practice, in *Proceedings of the Participatory Design Conference* CPSR, Palo Alto, CA pp 141–152

Flanagan, J C (1954) The critical incident technique *Mental Bulletin* Vol 51 pp 327–358

Gulliksen, J, Göransson, B, Boivie, I, Blomkvist, S, Persson, J and Cajander, Å (2003) Key principles for user-centred systems design *Behaviour and Information Technology* Vol 22 No 6 pp 397–410

Hesse, W (2003) Dinosaur meets archaeopteryx? or: is there an alternative for Rational's Unified Process? *Software and Systems Modeling* Vol 2 No 4 pp 240–247

Hull, M E C, Taylor, P S, Hanna, J R P and Millar, R J (2001) Software development processes – an assessment *Information and Software Technology* Vol 44 pp 1–12

Kensing, F. (2000) Participatory design in a commercial context – a conceptual framework, *Proceedings of the Participatory Design Conference* CPSR, Pao Alto, CA pp 116–126

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009

Kensing, F and Halskov Madsen, K (1991) Generating visions: future workshops and metaphorical design in **J Greenbaum and M Kyng** (eds) *Design at work* Lawrence Erlbaum, Hillsdale, NJ pp 155–168

Klein, G A, Calderwood, R and Macgregor, D (1989) Critical decision method for eliciting knowledge *Systems, Man and Cyberetics* Vol 19 No 3 pp 462–472

Kruchten, PH (2004) The Rational Unified Process. An introduction Addison-Wesley

Oesterreich, B (2002) Developing software with UML, 2nd edn. Addison-Wesley

Olsson, E (2004) What active users and designers contribute in the design process *Interacting with Computers* Vol 16 pp 377–401

Oostveen, A-M and van den Besselar, P (2004) From small scale to large scale user participation: a case study of participatory design in E-government systems, *Proceedings of the Participatory Design Conference* CPSR, Pao Alto, CA pp 173–182

Pilemalm, S (2002) Information technology for non-profit organisations. Extended participatory design of an information system for trade union shop stewards, Linköping Studies in Science and Technology Dissertation No 749

Reich, Y, Konda, S L, Levy, S N, Monarch, I A and Subrahmanian, E (1996)-Varieties and issues of participation and design *Design Studies* Vol 17 pp 165–180

Taylor, S J and Bogdan, R (1998) *Introduction to qualitative research methods* John Wiley and Sons, New York

Tollmar, K (2001) Towards CSCW design in the Scandinavian tradition department of numerical analysis and computer science, Stockholm University Doctoral Dissertation

Please cite this article in press as: Pilemalm, S et al., Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach, Design Studies (2007), doi:10.1016/j.destud.2007.02.009