Groupware Architecture for Requirements Processes in New Product Development

Ahti Salo
Systems Analysis Laboratory
Helsinki University of Technology
02015 HUT, Finland
ahti.salo@hut.fi

Timo Käkölä
Computer Science and Information Systems
University of Jyväskylä
40351 Jyväskylä, Finland
timo.kakola@cc.jyu.fi

Abstract

An organization's ability to create successful products depends on how extensively it collects, analyzes, and utilizes requirements information in its product development. Recent advances in groupware technologies allow such information to be shared more effectively in distributed organizations; thus, these technologies hold considerable potential as means of shortening the cycle time and improving the quality of requirements processes. This paper addresses issues in the design and implementation of groupware-supported requirements processes by drawing on experiences from large-scale industrial development projects within Nokia. The primary contribution of the paper is the conceptual design of a generic, groupware-based requirements management system (RMS) which is postulated to help product development organizations improve their requirements processes.

1. Introduction

The development of high technology products is characterized by pressures towards shorter time-to-markets, increasing complexity of product designs, globalization of markets, and continuous price erosion. To thrive under these conditions, it is crucial that the company not only helps its personnel become more knowledgeable about the marketplace but also ensures that the accumulated knowledge is fully leveraged [11, 15, 16, 19, 24]. This is particularly true during the earliest phases of new product development (NPD) during which different functional organizations need to integrate their expertise in order to develop a credible product definition [2]. Indeed, there is ample evidence that the synthesization of the knowledge possessed by the multiple stakeholders is one of the key determinants of successful product development [3].

In reality, however, the achievement of such an integration is complicated by several factors. The development activities in large companies are often scattered across multiple sites, which makes it hard to set up face-to-face meetings. At the same time, differences in incentives and organizational culture as well as divergent perceptions about the marketplace and the product's mission may stay in the way of reaching an agreement about the product definition [2]. In this context, the provision of communication support for the early phases of NPD activities thus poses a significant challenge with substantial payoffs.

The rapid progress of networking technologies over the past few years has offered new possibilities for the knowledge creation and sharing within organizations. Groupware support for co-ordination and collaboration, in particular, can be deployed to help organizations increase the productivity of knowledge-intensive work. It is the quest for these productivity gains which spurs multinational corporations to invest in networked information technology (IT) infrastructures on top of which knowledge-reliant processes may be run.

Nevertheless, there are only few studies that investigate (1) which processes within NPD are amenable to networked IT solutions, (2) what targets should be set for such solutions, and (3) how these targets can be translated into implementations. In their recent study of the intranet-enabled use of product specifications in a large NPD project, Hameri and Nihtilä [7] note that prior research on networked NPD-processes is very limited. Gorton, Hawryszkiewycz and Fung [6] report positive results from groupware support in their small scale experimental research, but are uncertain as to whether their findings will scale up into large development. Thus, there is a need for further work on the use of networked IT solutions for enhancing the effectiveness of industrial NPD processes in multi-site and multi-organizational settings.

The focus of this paper lies at the intersection of computer-supported collaborative work (CSCW) and requirements engineering (RE), the emphasis being on

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

how groupware solutions can respond to the needs arising from the requirements domain. Because NPD relies heavily on organizational memory (OM), we address that domain as well. Our discussion is based on the analysis of activities that typically take place during the early phases of NPD and, more importantly, on the translation of these activities into design considerations for groupware-based requirements management support.

We draw upon experiences gained during the migration of requirements management (RM) processes of several product lines within Nokia onto a groupware platform. Key people from R&D and product marketing were informally interviewed to address issues in the status-quo analysis, target setting, process redesign, development of groupware solutions and, more generally, the institutionalization of the new RM processes. The interviews were conducted and examined under heavy time pressure for supporting this *industrial* work and some sacrifices with respect to the methodological rigour had to be made. An extensive literature review was conducted to support the design work and the interpretation of the interviews. But the interviews could not be recorded and transcribed, and thus cannot be quoted in this paper. Rather, notes were taken from each interview and analyzed with respect to the above-mentioned issues. Because our experiences coincide well with the insights gained from the literature review, we believe that the methodology does provide a sound justification for the discussion in this paper. However, a detailed empirical study on the interactions between the groupware solutions and the new RM processes does remain an important topic for further research.

Instead of looking at any one of the product lines or groupware solutions in depth, we highlight commonalities which prevailed in the case of all products and, due to their generality, are likely to be relevant in other NPD contexts as well. Specifically, we focus on the structure of distributed requirements processes in terms of their properties and implications for the design of a generic groupware-based requirements management system (RMS). Much of the attention is given to the underlying principles which seem relevant for the enactment of similar processes on top of other IT platforms. The socio-technical requirements for designing effective groupware-supported requirements processes are covered in a related paper [20].

This paper proceeds as follows. Section 2 briefly covers the role of groupware in NPD. Section 3 outlines a conceptual design of RMS. Conclusions and issues for future research are discussed in the last section.

2. Groupware in new product development and requirements management

The groupware marketplace is currently one of the hottest areas of IT. Up until recently, however, much of groupware application development has focused on supporting administrative and operational processes - such as order processing and purchasing - that are structured and not particularly knowledge-intensive [12]. Due to technical advances in commercially available groupware platforms (e.g., improved robustness and usability), these platforms now provide viable options for the use of groupware in the redesign and implementation of product development processes which, by their very nature, are knowledge-intensive and ill-structured [5].

From the viewpoint of enhancing NPD effectiveness, it is pertinent to ask how groupware solutions and associated communication spaces should be crafted to support NPD processes. Since NPD processes are complex and cross-functional activities with long time-spans, redesigning them to leverage the potential of groupware is a strategic and resource-intensive effort. Little research is available to help organizations in the prioritization and execution of their NPD redesign efforts.

Our experiences at Nokia and survey of research literature suggest that requirements management is one of the areas of NPD where the deployment of groupware solutions can lead to significant process improvements. Specifically, RM processes are concerned with

- the systematic collection of information about customer needs, technical constraints and, more generally, any information which needs to be accounted for in product decisions,
- the refinement of such information into representations that are suitable for systematic evaluations within the NPD organization, and
- the preparation and recording of product decisions as part of the earliest milestone reviews in product development.

Several reasons suggest that requirements processes, perhaps more than other phases of NPD, are amenable to groupware support. First, the need to establish good communication and collaboration patterns between different functional groups is often highest in requirements capture and analysis [10]. Second, this phase - sometimes referred to as the “fuzzy front end” - frequently belongs to the largest and cheapest opportunities to shorten the development cycle [21]. Third, groupware solutions contribute to effective sharing and integration of knowledge, processes that must be supported by advanced software development environments [4]. Furthermore, earlier research indicates that many of the problems during the later phases of the

product life-cycle, particularly in terms of additional effort, are caused by fluctuating and conflicting requirements as well as communication and co-ordination breakdowns [4]. Taken together, these observations imply that groupware may be a promising platform for the institutionalization of requirements processes, particularly in large organizations with multi-site and cross-organizational development activities.

Likewise, RM processes pose challenges from the perspective of organizational memory (OM), that is, "the means by which knowledge from the past is brought to bear on present activities, thus resulting in higher or lower levels of organizational effectiveness" [22, p. 89]. We focus on challenges that are, from an epistemological viewpoint, primarily related to explicit, codified knowledge (hereafter "EOM") that can be processed with computers. Tacit knowledge rooted in human intentional action, while an important component of OM, is beyond our scope. The first of these challenges is that products are often delivered to customers as incremental releases to earlier versions of the product. As a result, it is critical that the development organization can retrieve up-to-date information from EOM about the features that have been supplied in earlier releases or, alternatively, are planned for implementation in future releases of the product. Second, to the extent that requirements and new product features stored in EOM contain information about the milestone plans and work effort estimates, the organization can systematically acquire an experiential basis for improving its project and release planning. Third, for several reasons (e.g., costs, personnel turnover), organizations should not be too dependent on immediate access (e.g., phone, face-to-face meetings) to the expertise of individual experts. EOM reduces this dependence, especially if it can be accessed with few restrictions on time and place. Fourth, EOM is crucial for sharing knowledge across organizational levels, product lines, and functions involved in NPD [14, 15, 25].

Groupware-based RMS is only one of the available options for supporting the creation and maintenance of EOM. In our case at Nokia, the motivation for choosing RMS and focusing on the earliest phases of requirements processes arose from the perceived need to establish more effective and transparent communication patterns between product marketing and R&D, on one hand, and from the availability of a global groupware infrastructure, on the other hand. In this situation, the available groupware infrastructure was seen as a potential medium for establishing more effective communication patterns among the stakeholders. This is not to say that other processes in NPD (e.g., detailed design or system testing) would not have been suitable for groupware support; rather, it was felt that the payoff of such support would be higher in the context of requirements processes, given the geographical dispersion and extensive communication needs of stakeholders.

The main objective of the groupware solutions at Nokia was to implement robust, scaleable, and effective processes for the activities which precede the earliest milestone reviews. These activities include:

- the capture of new product ideas from product marketing, R&D, and other relevant sources as an input to a structured requirements process;
- the conversion of such ideas into well-defined expressions of product functionality;
- the assessment of suggested functionalities in view of their market potential, technical feasibility, compliance with product strategy, and other factors which bear on product decisions;
- support for and recording of go/no-go decisions about the product functionalities;
- systematic monitoring of the process by all stakeholders; and
- the preservation and dissemination of all the history data resulting from the exchange of ideas on the electronic media.

The development of RMS was carried out jointly by Nokia Research Center, the corporate research unit of Nokia, and representatives from the business units. The activities that take place during the early phases of requirements management were analyzed to develop the early versions of RMS. Among other things, this analysis consisted of a study of the existing document structure and interviews with representatives from marketing and R&D. The ideas that emerged were instrumental in steering the development of early RMS pilot implementations. Once positive results had been made with these, the key elements of RMS design were largely frozen and RMS instances were rolled-out into full-scale operational use. The adoption of RMS instances proceeded in parallel with the build-up of a corporate-wide groupware infrastructure.

The groupware solutions referred to in this paper were implemented on top of Lotus Notes™ (see, e.g. [12]). However, we believe that our results, particularly the conceptual design of the generic requirements management system, can be generalized to other platforms with sufficiently flexible mechanisms for implementing role-oriented workflow processes.

3. Implementation of groupware support

This section outlines a groupware architecture for requirements management. The architecture presented is by no means the "best" or the only one. Rather, it exemplifies a conceptual design that was found generic and applicable in the context of Nokia. This architecture has been translated into several instantiations of a Lotus

Notes™-based Requirements Management System (RMS) at different product lines of Nokia. Thus, when we refer to experiences with specific design aspects of groupware-based requirements processes or their impact, we draw on experiences from the operational use of these RMS applications.

3.1. Phases of a requirements process

Here, associated with the work of Yeh [25], the processes supported by RMS include the following phases of requirements management:

Table 1. Phases in requirements management

Phase	*Description*
Capture	The process of collecting new product ideas and requirements from relevant sources with minimal control on representation or content.
Categorization	The association of submitted requirements with appropriate context.
Refinement	The transformation of requirements information into units about which a tentative product decisions can be made.
Assessment	The evaluation of requirements information by relevant stakeholders in the light of endorsed targets and decision criteria.
Follow-up	Systematic monitoring of requirements usage during the later phases of the product development process.

Since RMS has a central role in supporting the organizational memory of requirements management and NPD processes, it is important to associate these five phases with the attendant processes of memory, that is, acquisition, retention, maintenance, search, and retrieval of information [22, 24]. Requirements capture and categorization primarily relate, respectively, to the acquisition and retention of information. Requirements refinement and assessment relate mainly to the maintenance of information. Finally, requirements follow-up deals mainly with the search and retrieval of information.

3.1.1. Requirements capture. In this phase, RMS is to facilitate the distributed creation and collection of requirements from a wide audience within the corporation. To foster innovativeness and acceptance of RMS within the organization, excessive constraints on the representation and substance of requirements should be avoided at this stage. From a practical perspective, the key issue is that of making RMS easily accessible to all stakeholders, for example, by giving them the possibility to submit new ideas at the time when they first emerge.

3.1.2. Requirements categorization. Submitted requirements need to be placed into their proper context to support retention, retrieval, interpretation, clarification and validation. The customer's use environment is an important contextual anchor [2] and can be accommodated, for instance, by linking the requirements to the customer's business processes and other specific customer characteristics. More generally, contextualization can be supported by associating requirements with the overall product architecture and the NPD organization, including specific roles and responsibilities. The design of RMS features for this phase is presented in Section 3.3.

3.1.3. Requirements refinement. The raw material created in the earlier phases must be converted into well-defined concepts, i.e. prospective product functionalities, about which product decisions can be made. This activity takes place through three intertwined processes:

1. *requirements aggregation* where small entities are assembled into larger concepts,
2. *elimination of redundant concepts* where similar ideas (which may be invented by several people in a large organization) are linked to each other so as to avoid duplication of effort in their later analysis,
3. *requirements separation* where large concepts are divided into smaller ones in order to derive product functionalities about which separate decisions need to be made. In the case of system deliveries, for instance, these decisions include the allocation of related requirements to different product lines.

The central RMS features for these processes include traceability support and layering of documentation. The design of RMS features for this phase is presented in Section 3.2.

3.1.4. Requirements assessment. In this phase, requirements and proposed new product functionalities are evaluated to ensure that they comply with the overall product strategy as well as budget and other resource constraints. To achieve this, product decisions must be of a high quality, that is, they need to be based on best available information and taken by the right people in a timely, consistent, comprehensive, justifiable and identifiable manner so that one can, at a later time, determine where, when, and by whom the decision was taken. The design of RMS support for this phase is presented in Section 3.5.

3.1.5. Requirements follow-up. Once the product decision has been made, the approved requirements provide a foundation for the later phases of product development. The typical phases in the context of Nokia are the specification, implementation and testing of the product. The requirements contained in the RMS must be made visible to the entire organization involved in these phases. Aspects supporting follow-up processes are covered in Sections 3.2 and 3.4.

3.2. Layering of documentation

For the purpose of separating the inputs to a requirements process from their implications in terms of new product functionalities, it is helpful to establish a layered documentation structure in which the initial expressions of customer needs and other requirements are separated from their implications for product design. Layering not only permits the application of separate quality controls to the different layers, but also provides support for deriving and refining product functionalities from a continuous influx of ideas.

In its simplest form, the layered documentation structure consists of two layers only. In order to stress the difference between these layers, we have chosen *product idea* and *feature proposal* as labels for referring to the two forms of requirements information (see Table 2). Conceptually, the adoption of these terms helps to distinguish observed customer needs and corresponding product functionalities, much in the same way as the terms "user requirement" and "system requirement" in structured requirements analysis (and associated commercial software packages; see Stevens [23]) are used to separate customers' concerns from technical considerations.

Table 2. Two-layered document structure

Document	***Definition***
Product idea	An expression of a customer need or some other desired property that may call for the implementation of a new or enhanced product functionality.
Feature proposal	A description of a product functionality, expressed in a detailed and unambiguous format to permit the making of a go/no-go decision.

The two-layered documentation structure supports the channeling of product ideas into suggestions about new product features so that go/no-go decisions about the launch of feasibility studies can be taken. Extending the scope of groupware support beyond feasibility studies would motivate the introduction of further documentation layers. Additional documentation layers might also be called for in situations where the product ideas are very dissimilar in terms of their size and scope. In such a situation, the intermediate layers would assist in the conversion of broad concepts into their constituent product functionalities and, conversely, in the aggregation of small enhancements into more extensive product functionalities.

One of the reasons for the layered documentation structure is that in a large development organization, the requirements expressed by different customers often resemble each other and several people may come up with similar ideas for improving the product. Thus, the conversion of every new product idea into a full-fledged feature proposal would entail a considerable and partly duplicate effort. Redundant work can be reduced or avoided by associating the ideas that seem similar or closely related with a single feature proposal. The proposal and all the associations can then be assessed as a whole. In other words, the rationale for creating links between product ideas and feature proposals stems, in part, from the need to implement an efficient assessment process.

Bi-directional links between product ideas and feature proposals are critical because they provide full traceability support. These links permit the originators of product ideas to examine the feature proposals that these suggestions have inspired, including the status of these proposals in the development process; at the same time, the links are valuable as designers can access and study the underlying ideas behind the feature proposals. In fact, one of the greatest advantages of these links may be that designers can easily get in touch with the originators of the ideas. Improved communication between designers and users (and their "representatives", primarily marketing) reduces the time-space disjuncture between the design and use of technology, thus contributing to the design quality and the effective use of technology [17].

Based on our experience with RMS, the number of incoming ideas may be several times (e.g., three to five) larger than that of recorded feature proposals which, again, is indicative of the need to establish a layered documentation structure. The links between product ideas and feature proposals may be many-to-many, and they are maintained primarily by the persons who are responsible for the feature proposals. In practice, new product ideas are first associated with the product architecture, which is then used to assign further responsibilities in the analysis of new product ideas and derived feature proposals. This practice relieves the contributors of new product ideas from the need to attend to the further processing of the submitted ideas, even if they may receive requests for

clarification later on. It also helps in keeping the traceability links in RMS up-to-date, which is critical for retaining the organization's trust in the validity and timeliness of the RMS database.

3.3. Contextual relationships and context modeling

To facilitate their retention and later search and retrieval, product ideas need to be associated with their proper context within the product architecture and the corresponding organizational responsibilities. For the sake of clarity, it is helpful to divide the information on requirements documents into separate sections which provide answers to specific questions. Table 3 presents such a division and some examples of related attributes.

For *product ideas*, the origins of the requirement must be fully specified. For *feature proposals* derived from product ideas, the section for origins may even be omitted since the information can be identified from the corresponding product ideas through the use of the traceability links. The analysis section, on the other hand, should be completed with due care.

An RMS instantiation can be derived by choosing appropriate sections from the above template, whereby the domain models associated with the product structure and the NPD organization also need to be developed. Moreover, in order to manage the requirements workflow, it is necessary to choose (1) a set of status flags that correspond to the different phases in the processing of requirements (e.g., *New - Categorized - Analyzed - For Review - Approved / Rejected / Postponed)* and (2) the persons to whom the requirement may be assigned in these phases (e.g., on the basis of particular expertise associated with certain customers or parts of the product).

3.3.1. Role- or phase-sensitive document layouts. An important element in context modeling can be that of changing the layout of the document depending on the particular phase the document is in. For instance, information about the due date of a review and reviewers might be placed at the top of each document that is currently under review so that this information – being critical to the further processing of the document - would immediately catch the eye of the reviewer.

However, context-sensitivity needs to be employed with caution for two reasons. First, implementing a context-sensitive layout may complicate the process and consequently the design of RMS as well, making it harder to adapt the RMS to any future changes. Second, context-sensitivity may confound users unless it is implemented in an intuitive fashion, because otherwise users - particularly those who do not use the RMS regularly - may experience difficulty in understanding "what is going on" or why the system behaves the way it does.

3.3.2. History. In terms of traceability and information control, a critical element in requirements management is that of maintaining organizational memory of how, when and by whom the requirements have been modified. Towards this end, the requirements can be supplemented with a history section which accumulates triplets consisting of (1) the date of modification, (2) the name of the person modifying the document, and (3) the status of the document in the process (e.g., *New, Rejected, Approved* etc.). By examining these, the users can get an overview of the requirement's earlier phases and, if need be, even contact some of the persons who have introduced these modifications (c.f., [18, pp. 73-75]).

Table 3. The generic structure of a requirement

Class	***Question***	***Attributes***
Description	What is the requirement about?	Description Rationale
Origin	Where does the requirement come from?	Author Source Date of creation
Categori-zation	What parts of the product and the development organization is the requirement related to?	Traceability links Position in product structure (c.f. archi-tecture) and asso-ciated organizatio-nal responsibilities Interfaces to other product lines
Analysis	What are the implications of the requirement?	Status Priority Customer need Required work effort Risks
Workflow	What should be done to this requirement next? By whom?	Task description Assignments to persons
History	What has been done to the requirement? When?	Information about all prior edits, editors, and changes

3.4. Information navigation

In Lotus Notes™ databases, information search and retrieval are supported through views which, in essence, are document lists where one or more of the leftmost

columns are sorted or categorized. In this way, views support the rapid identification of the documents that are being looked for, provided that there is a view in which the categorizations match those document attributes along which the search is being made. Views combined with the ability to store multiple types of information in documents provide a powerful support for organizational memory [13, pp. 263-264].

To make the most out of views and the associated possibilities for navigation, it is helpful to identify and exploit the different dimensions along which requirements related information may be searched for. At the broadest level of analysis, it is possible to identify role-, content-, and process-related views, of which typical examples are given in Table 4.

In large development organizations, it may be desirable to limit the visibility of some requirements to a subset of the NPD organization. Such limitations reduce, for instance, the possibility of information leaks or premature conclusions by marketing staff about the features which might be forthcoming in future releases of the product. In this case, it is necessary to manage read and write access rights in RMS at the level of individual documents, whereby the responsible authors are put in charge of determining the persons (or groups thereof) who are entitled to read or modify requirements.

Table 4. Role, content, and process dimensions for information navigation

Axis of orientation	***Basis of information organization (Examples in italics)***
Role	The role in which the user interacts with the requirements database. *(e.g., Requirements sorted by their author)*
Content	Content-related assessments which have been made about the requirement. *(e.g., Requirements sorted by their priority)*
Process	Phase in the document life-cycle. *(e.g., Requirements sorted by their review status)*

In addition to standard views (see Table 4), two other features can be extremely helpful in supporting navigation with RMS: *To Do*-views and email notifications. *To Do*-views (or "private views" in the Notes™ terminology) would contain, from the perspective of each individual user, those documents that are specifically under his or her responsibility, or alternatively, assigned to him or her for further analysis and response. Such views make the application easily approachable by end-users, as most of the tasks can be readily completed through the use of these views only.

To ensure that people will indeed notice and react to their assignments, email notifications may be called for. In our RMS, notifications were sent to those persons from whom additional inputs and reactions were solicited. They were instrumental in gaining the attention of the experts from whom inputs were being solicited on a more occasional basis.

3.5. Finding right evaluation practices and right people for requirements assessment

Formal evaluation approaches such as QFD [1, 8, 9] can contribute to requirements assessment as they force stakeholders (potentially including the end-users of future products) to examine how the individual requirements contribute to the objectives that have been placed on product development. Thus, a central RMS design issue is what types of assessment and decision support facilities should be enabled by RMS.

In a distributed organization, with a high-volume and continuous incoming stream of new requirements, the adoption of formal and often more complex evaluation procedures can pose difficulties. On one hand, the sheer amount of new data limits possibilities for using any complicated procedures; on the other hand, the possibilities for organising meetings are limited, wherefore more "lightweight approaches" to decision support may be preferred.

Consequently, the RMS design was developed so as not to impose formal evaluation practices. Instead, a separate review form was provided on which the most significant concerns were listed (e.g., usability, market value, and technical feasibility). The reviewers could then use this form to supply their verbal judgements on the matter at stake. In this way, the richness and flexibility of verbal communication was relied on in a search for a balance between

- the complexity of the method,
- the amount of information that needs to be evaluated, and
- the effort which can be realistically devoted to the analysis.

Since in a large distributed organization it is often difficult to identify and reach the right persons who should analyze and examine the requirements, RMS was coupled with an organizational model that explicated the competencies associated with certain fields of expertise. This was done through the implementation of adjunct databases containing information about the persons who were specialists in, say, specific market areas or technical standards. With the help of this additional information, the requirements could be allocated to the most competent persons for review. The conceptual design of RMS thus

provided a way of the modeling of "know-who", which was found to contribute to the visibility of knowledge and the extensibility of the system.

4. Conclusions and future research

This paper has focused on the design and implementation of groupware-supported requirements processes in the light of experiences from industrial development projects at Nokia. Its primary contribution is the conceptual design of groupware functionalities which support the early phases of requirements processes in large-scale multi-site NPD organisations. Since these processes have been successfully transformed in several product lines within Nokia (with hundreds of active RMS users), we expect that our design is of relevance to other product-development organizations.

While careful empirical work lies beyond the scope of this paper, there is nevertheless a need to collect and analyze information on how RMS usage affects the working and learning practices of product developers and entire development organizations. Hence, a research agenda is needed to investigate:

1. To what extent RMS instances following the design considerations in this paper are generic enough to respond to the needs of NPD organizations?
2. How the conceptual design of RMS might be enhanced to make it more generic and more detailed, thus increasing its relevance and applicability in industrial contexts?
3. Which changes in organizational structures and NPD processes may be necessary to leverage the potential afforded by RMS?

A limitation of this paper is that the implicit worldview (c.f., [13]) of the authors (as designers and researchers) has been significantly shaped by Nokia and its Notes-based groupware architecture. As a result, this paper may have emphasized some design aspects of RMS while downplaying or neglecting other equally or even more important aspects. For example, groupware support for synchronous remote interactions might be useful in some phases of requirements management, but it has been beyond the scope of the design of RMS, partly because the groupware architecture we have relied on does not support such interactions.

Addressing the above-mentioned research agenda would transcend this limitation and contribute to an enhanced understanding of groupware-supported NPD activities, thus helping organisations in the design of effective, generic requirements management processes. There are at least three levels that such an agenda may span:

1. the team level - for understanding how teams shape and are shaped by RMS;
2. the product line level - for understanding how RMS mediates the sharing of knowledge between different teams over time within a product line;
3. the organisational level - for understanding how RMS mediates the sharing of knowledge between different product lines over time.

The research agenda can be further structured by considering to what extent RMS has - in comparison to the situation before its introduction -

1. increased the depth of analysis;
2. improved the traceability of requirements;
3. shortened the time of developing new product concepts;
4. facilitated the creation of shared knowledge about technological opportunities and constraints as well as new market opportunities; and
5. facilitated the establishment of incentive schemes that foster shared knowledge creation within and between product development teams and across functions and product lines.

Each of the above questions need to be addressed to obtain guidance for the future development of RMSs. It is expected that - as Burchill and Fine [2] note - the more often organizations can answer them affirmatively, the more effective their NPD processes are likely to be in terms of requirement clarity, design objective credibility, work force commitment, and reduced cycle time. Moreover, answering these questions will extend the current NPD literature towards a deeper appreciation of the possibilities afforded by modern communication technologies.

5. References

1. Akao, Y., (Ed.) *Quality Function Deployment.* (1990), Productivity Press: Cambridge, MA.
2. Burchill, G., and Fine, C. H. *Time Versus Market Orientation in Product Concept Development: Empirically-Based Theory Generation.* Management Science *43*, 4 (1997), 465-478.
3. Cooper, R. G., and Kleinschmidt, E. J. *What separates winners from losers?* Journal of Product Innovation Management *4*, 3 (1987), 169-184.
4. Curtis, B., Krasner, H., and Iscoe, N. *A Field Study of the Software Design Process for Large Scale systems.* Communications of the ACM *31*, 11 (1988), 1268-1287.
5. El Sawy, O. *Minding Your Own Business Processes: The BPR Learning Book.* McGraw-Hill, (in press).
6. Gorton, I., Hawryszkiewycz, I., and Fung, L. *Enabling Software Shift Work with Groupware: A Case Study.* In Proceedings of the *29th Annual Hawaii International Conference on System Sciences* (Maui, Hawaii, USA, 1996), IEEE Computer Society Press, Los Alamitos, CA.
7. Hameri, A. P., and Nihtilä, J. *Distributed New Product Development Project Based on Internet and World-Wide Web: A Case Study.* Journal of Product Innovation Management *14*, 2 (1997), 77-87.

8. Hauser, J. R., and Clausing, D. *The House of Quality*. Harvard Business Review *66*, 3 (1988), 63-73.
9. Hrones, J. A. J., Jedrey, B. C., and Zaaf, D. *Defining Global Requirements with Distributed QFD.* Digital Technical Journal *5*, 4 (1992), 36-46.
10. Katz, R., and Tushman, M. *Communication Patterns, Project Performance, and Task Characteristics - An Empirical Evaluation and Integration in an R&D Setting*. Organizational Behavior and Human Performance *23*, (1979), 139-162.
11. Leonard-Barton, D. *Wellsprings of knowledge: building and sustaining the sources of innovation.* Harvard Business School Press, Boston, MA (1995).
12. Lloyd, P., and Whitehead, R. *Transforming Organizations Through Groupware: Lotus Notes in Action.* Springer, (1996).
13. Mandviwalla, M., and Olfman, L. *What Do Groups Need? A Proposed Set of Generic Groupware Requirements.* ACM Transactions on Computer-Human Interaction *1*, 3 (1994), 245-268.
14. Nonaka, I. *The Knowledge-Creating Company*. Harvard Business Review, November-December (1991), 96-104.
15. Nonaka, I. *A Dynamic Theory of Organizational Knowledge Creation.* Organization Science *5*, 1 (1994), 14-37.
16. Nonaka, I., and Takeuchi, H. *The Knowledge-Creating Company*. Oxford University Press, New York (1995).
17. Orlikowski, W. *The Duality of Technology: Rethinking the Concept of Technology in Organizations.* Organization Science *3*, 3 (1992), 398-427.
18. Orlikowski, W. *Improvising Organizational Transformation Over Time: A Situated Change Perspective.* Information Systems Research *7*, 1 (1996), 63-92.
19. Quinn, J. B., Anderson, P., and Finkelstein, S. *Managing Professional Intelligent: Making Most of the Best.* Harvard Business Review, March (1996), 71-80.
20. Salo, A., and Käkölä, T. *Requirements for Groupware-Supported Requirements Processes in New Product Development.* In Proceedings of the *Fourth International Workshop on Requirements Engineering* (Pisa, Italy, 1998), Springer.
21. Smith, P. G., and Reinertsen, D. G. *Shortening the Product Development Cycle. Research Technology Management 35*, 3 (1992), 45-49.
22. Stein, E. W., and Zwass, V. *Actualizing Organizational Memory with Information Systems.* Information Systems Research *6*, 2 (1995), 85-117.
23. Stevens, R. *Structured Requirements.* In Proceedings of the *Fourth International Symposium on Systems Engineering* (1994), National Council on Systems Engineering, Sunnyvale, CA, 99-104.
24. Walsh, J. P., and Ungson, G. R. *Organizational Memory*. Academy of Management Review *16*, 1 (1991), 57-91.
25. Yeh, A. C. *REQUirements Engineering Support Technique (REQUEST): A Market Driven Requirements Management Process.* In Proceedings of the *Second Symposium on Assessment of Quality Software Development Tools* (1992), IEEE Computer Society Press, Los Alamitos, CA, 211-223.