IGI PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.igi-global.com

ITJ4028

This paper appears in the publication, **Journal of Database Management**, Volume 19, Issue 1
edited by **Keng Siau © 2008, IGI Global**

Conflicts, Compromises, and Political Decisions:
Methodological Challenges of Enterprise-Wide E-Business Architecture Creation

Kari Smolander, Lappeenranta University of Technology, Finland

Matti Rossi, Helsinki School of Economics, Finland

ABSTRACT

This article describes the architecture development process in an international ICT company, which is building a comprehensive e-business system for its customers. The implementation includes the integration of data and legacy systems from independent business units and the construction of a uniform Web-based customer interface. We followed the early process of architecture analysis and definition over a year. The research focuses on the creation of e-business architecture and observes that instead of guided by a prescribed method, the architecture emerges through somewhat non-deliberate actions obliged by the situation and its constraints, conflicts, compromises, and political decisions. The interview-based qualitative data is analyzed using grounded theory and a coherent story explaining the situation and its forces is extracted. Conclusions are drawn from the observations and possibilities and weaknesses of the support that UML and RUP provide for the process are pointed out.

Keywords: *architecture; e-business; enterprise; IS architecture; IS development methodologies; IS integration; RUP; UML*

INTRODUCTION

Robust technical architecture is considered one of the key issues when building successful e-business systems. The design of technical architecture is usually seen as a set of trade-offs between available resources (such as available personnel and money) and operational requirements related to technical architecture, such as scalability, capacity, response times, security, and availability. The software architecture research provides design tools for technical architecture design, including, for instance, architecture description languages (Dashofy, Van der Hoek, & Taylor, 2005; Medvidovic & Taylor, 2000), common architectural patterns and styles (Monroe, Kompanek, Melton, & Garlan, 1997), architectural trade-off methods (Kazman, Klein, & Clements, 2000), architectural frameworks (Leist & Zellner, 2006), and technologies for e-business implementation (Bichler, Segev,

& Zhao, 1998). In an ideal world, the work of an architect would be to find the explicit requirements for architecture, and select the best possible design tools and technologies to implement the architecture. Furthermore, the architecture development team would make rational trade-offs concerning the requirements, and produce the best realistic solution for the architecture with the selected design tools and implementation technologies.

However, the literature contains many examples of cases where technical rationality has not been sufficient for the success in IS projects (e.g. Sauer, Southon, & Dampney, 1997). Architecture researchers have found that the work of an architect and the usage of architecture are bound by more diverse organizational issues and limitations that the classical technical software architecture research ignores. These include for example the diverse role of an architect in an organization observed by Grinter (1999) and varying uses and meanings of architecture in practice (Smolander & Päivärinta, 2002a). The main message of these studies is that an architect has a social, and even political, role in an organization and that different stakeholders relate different meanings to architecture to fulfill their informational requirements in the development process. This phenomenon has remarkable similarities to information systems development in general. As pointed out by Klein & Hirscheim, the implicit assumption of rationality of the development processes hides the legitimating of the goals and differing political agendas of various stakeholders (Hirschheim & Klein, 1989).

To understand the issues involved in architecture development, we observed a project that was developing e-business architecture in an international ICT company. We interviewed various stakeholders to gain a deep insight into the process. The company already had several e-commerce systems in individual business units, but it needed a more uniform customer interface for its various systems. The e-business project included the integration of data and legacy systems from these units and the construction of a uniform Web-based customer interface hiding the differences of the business units. Our goal was to find ways for supporting architecture development by means of methods and description languages, such as UML. We were aware of efforts of supporting architecture design with UML (e.g., Conallen, 1999; Garlan & Kompanek, 2000; Hofmeister, Nord, & Soni, 1999b; Object Management Group, 1999, 2006), but these efforts were mostly targeted to technical software design and we did not know how well these would support a large socio-technical or organizational project, such as enterprise or e-business architecture development. Therefore we decided to observe a real world project and concentrate on the requirements that e-business architecture development in its complex organizational context state on description languages and development methods. Next, we decided to compare the observed requirements to the support that UML and RUP offer, because they, together, form the current methodological basis for many systems development organizations. UML is the de-facto standard language in software and systems development and RUP (Jacobson, Booch, & Rumbaugh, 1999) is a widely known process model that claims to improve development process maturity (Kuntzmann & Kruchten, 2003). We believed that this kind of knowledge would benefit both practitioners in process improvement and developers of UML extensions.

Another interest was to find out what factors influenced the creation of e-business architecture: was it designed purposefully by software architects through rational decisions and trade-offs, or did it emerge through somewhat non-deliberate actions obliged by the situation and its constraints, conflicts, compromises, and political decisions? This is a very important issue, as unlike software architecture, e-business architecture is very tightly coupled with the business models of the company and thus the architecture has a far more direct impact on business than for example low-level system architecture. Furthermore, if the business models are not supported by the e-business architecture,

then the business strategy will not work (Ross, Weill, & Robertson, 2006).

We used open interviews of various actors in the projects to gather the necessary information about the project. We analyzed the qualitative data from the interviews using grounded theory (Glaser & Strauss, 1967) as the research method and concluded the analysis by categorizing the issues that had emerged using the taxonomy of Lyytinen (1987). Thus, we classified the issues as belonging into technical, language and organizational context. From this classification of issues, we extracted requirements for development methods when developing integrated e-business solutions and compared these requirements to the support that the combination of UML and RUP provides.

We observed that most of the problems encountered had very little to do with descriptions of the architecture per se. Rather what was problematic were the issues that architecture development exposed about the underlying organization. This is an important finding, as most of the research into architecture has been about effective description languages and design processes and there is a void of research about the organizational consequences of architecture development.

The article is organized as follows: we start by explaining in more detail what is meant by architecture in this article (section 2). In section 3, we describe the research process and method used. section 4 describes the situation the company is facing and the motives for the change and implementation of the e-business system. In section 5, we describe the situation and the context of the development project aiming at e-business implementation and the consequences of the situation for the progress of the development project. From the observed issues faced by the development project we draw conclusions and extract the requirements for development methods in e-business architecture development and compare the requirements to support that the combination of UML and RUP provides (section 6). We point out areas where current research is not supporting the needs of the practice of general and particularly e-business architecture development.

ARCHITECTURE IN SYSTEMS DEVELOPMENT

In this study, we describe a process where comprehensive e-business architecture is being created. In addition to e-commerce systems serving external customer transactions, e-business includes both the integration of and streamlining of internal information systems to serve the new digitally enabled business processes (Kalakota & Robinson, 2001) and the unified customer interface (Ross et al., 2006). For the sake of simplicity, we understand e-business here to cover both the transactions and processes within a firm and the integrated external e-commerce systems as in (Kalakota & Robinson, 2001). This enables us to interpret the process in the studied organization as the process of building an integrated e-business architecture. Ross et al. (2006) stress the architecture as the necessary foundation for execution of comprehensive, across the functions operating, e-business.

Conventionally, architecture is understood as a high-level logical abstraction of the system defining the main components of the system and their relationships. The term architecture is also used both in the context of an individual system and in the context of systems integration. The software architecture typically concentrates on the architecture of a single software system, whereas the terms information systems (IS) architecture and enterprise architecture (Kim & Everest, 1994; Ross et al., 2006; Sowa & Zachman, 1992) refer to the overall architecture of all information systems in an organization.

In practice, however, the borderline between a single system and a set of systems is difficult to determine. Practically no system today is isolated from other systems, and the relationship of a system to its environment may be architecturally more important than the inner structure of the system, especially when developing e-business systems. Usually, systems rely on a common technical infrastructure, (including networks, processing services, operation services, etc.) which is common for

all the systems in an organization. Organizationally, architecture design is a co-operative effort involving many roles in the development environment. These roles include the role of an architect who is specifically associated with the task of architecture design. An architect needs contribution and commitment from many individuals, teams, and parts of organization to succeed in the effort (Grinter, 1999).

By architecture development, we mean a process where early design decisions are realized into an architecture defining that defines system's composition from various viewpoints. Architecture also contains the blueprints for system's implementation from conceptual and physical components. This process forms a set of documents which different stakeholders can use to relate their concerns to the issues made concrete by the architecture and discuss their needs in the terms defined by the common architecture. They can also make decisions concerning system development strategies and policies using architecture as a common reference. This conception sees architecture not only as a technical artifact but also as a boundary object (Star & Griesemer, 1989) having strong organizational connotations.

The conventional role of architecture is to serve as an enabler for further design and implementation (Hofmeister, Nord, & Soni, 1999a; Shaw & Garlan, 1996). Obviously, sound and well-designed technical architecture makes the detailed design and implementation of a system easier and less risky than it would be without such architecture. Architecture defines, for example, the modules or components which the system is composed of, and therefore it focuses and constrains the solution space of individual designers that develop individual components. This technical view of architecture has produced also studies related to UML. In the end of last decade, possibilities and weaknesses of UML as an architecture description language, and its complexity (Siau & Cao, 2001; Siau, Erickson, & Lee, 2005) were widely evaluated and enhancements were proposed (Conallen, 1999; D'Souza & Wills, 1998; Egyed & Medvidovic, 1999; Garlan & Kompanek, 2000; Hofmeister et al., 1999b; Medvidovic, Egyed, & Rosenblum, 1999; Rumpe, Schoenmakers, Radermacher, & Schürr, 1999). The recent developments in this area include the SysML extension of UML (Object Management Group, 2006). Different profiles and enhancements to UML have been proposed to tackle its limitations in electronic commerce (Dori, 2001).

RESEARCH PROCESS

The studied organization is a globally operating ICT company having thousands of employees worldwide. Its customers include both consumers and businesses for which the organization provides various products and services. Software is one of the key assets in the organization's service production and product development. Historically, the organization has had several independent business units targeted at diverging business sectors. In addition, the information management of the organization has been distributed to these business units and the functions of enterprise level information management have included mainly the provision of network infrastructure, enterprise level accounting, and basic office tools. Most of the information systems in use have been implemented and operated by the business units that have been quite independent in their decisions concerning strategies for information management. However, recent developments in markets and technology have led the organization to set its strategies to a more integrative direction. For this reason, the organization has set an objective to provide an integrated e-business solution to both its consumer and business customers. This will include both implementation of a uniform Web-based customer interface and sufficient integration between the distributed operative back-end information systems, such as customer management and billing systems.

The research process followed the grounded theory method (Glaser & Strauss, 1967), which is a research method developed originally for social sciences by Glaser and Strauss in the 1960s and later developed and reinterpreted by the original authors (e.g., Glaser, 1978; Strauss & Corbin, 1990) and others (e.g.,

Locke, 2001; Martin & Turner, 1986). Grounded theory promotes inductive theory creation from the data. The objective is not to validate or test theories but to create one. The analysis process of the grounded theory is explicitly defined and consists of several coding phases. The coding starts from *open coding* in which any incident, slice, or element of the data may be given a conceptual label for the identification of commonalities. These commonalities are called *categories* and they are described in terms of their properties (Fernández, Lehmann, & Underwood, 2002). The coding continues with *axial coding* (Strauss & Corbin, 1990) or theoretical coding (Glaser, 1978), where relationships between the categories are resolved. The coding ends at *selective coding* (Strauss & Corbin, 1990) where the resulting theory is "densified" (Glaser, 1978) or a core category selected (Strauss & Corbin, 1990) and theory about that is described. The data collection is based on the notion of *theoretical sampling*, which means adjusting the data collection process according to the requirements of the emerging theory. The sources of data may be adjusted during the process and the data collection can be stopped whenever a state of *theoretical saturation* is achieved, meaning a situation where no additional data would further develop the categories and their properties.

In the study, we interviewed 19 participants of the ongoing e-business system architecture design project during 2002, first in January and February and then later in November and December. The interviewees included six system architects, five enterprise system managers, three project managers, two software development managers, one project leader, one system analyst, and one marketing manager. Table 1 describes their relationship to the e-business development project. The interviews lasted from 45 to 120 minutes and they were completely transcribed as text.

The interview themes of this study were adjusted during the data collection to reflect better the developing theoretical understanding of the researchers and the specific knowledge of the interviewees. The emphasis of the interviews changed according to the interviewee and the special knowledge in his or her possession. Because the data collection proceeded partly in parallel with the analysis, the emerging theory

Table 1. Interviewed persons and their roles

Role	Tasks	Interviews
System architect	Deals with technological solutions and architectural structures in the e-business development project	6
Enterprise system manager	Is responsible for a portfolio of systems and technologies that are used in a particular organization. Acts as a customer in the internal e-business development project or participates it as an expert.	5
Project manager	Manages resources and is responsible for the execution of a sub-project of the e-business development project	3
Software development manager	Is responsible for a permanent software development organization	2
Project leader	Manages the e-business development super-project and supervises its set of sub-projects.	1
System analyst	Participates the requirements gathering and analysis phases as an intermediate between customers and technical experts.	1
Marketing manager	Is responsible for the public image and services of the electronic channel. Requirements setter and a customer to the development project.	1

also caused changes in the emphasis of the interview themes. In grounded theory this kind of adaptation is called *theoretical sensitivity*, and for theory-building research this is considered legitimate because "investigators are trying to understand each case individually and in as much depth as feasible" (Eisenhardt, 1989, p. 539). Eisenhardt calls the process where the emergence of a new line of thinking causes the altering of data collection *controlled opportunism* "in which researchers take advantage of the uniqueness of a specific case and the emergence of new themes to improve resultant theory" (Eisenhardt, 1989, p. 539).

The analysis in this study started with the open coding phase. In the beginning, we did not have any explicit *a priori* constructs for the analysis. Our task was to search mentions from the interviews that could be interpreted as meaningful related to the research question, "What are the conditions and constraints for creating and designing architecture in a large information systems development project?" The identified mentions related to this question were categorized using the software tool ATLAS.ti. During the open coding phase, altogether 187 emergent categories were found, and the categories were assigned to emerging scheme of super categories or category families, including for instance changes, conflicts, consequences, experiences, problems, purposes, and solutions occurring during the e-business architecture design and implementation process.

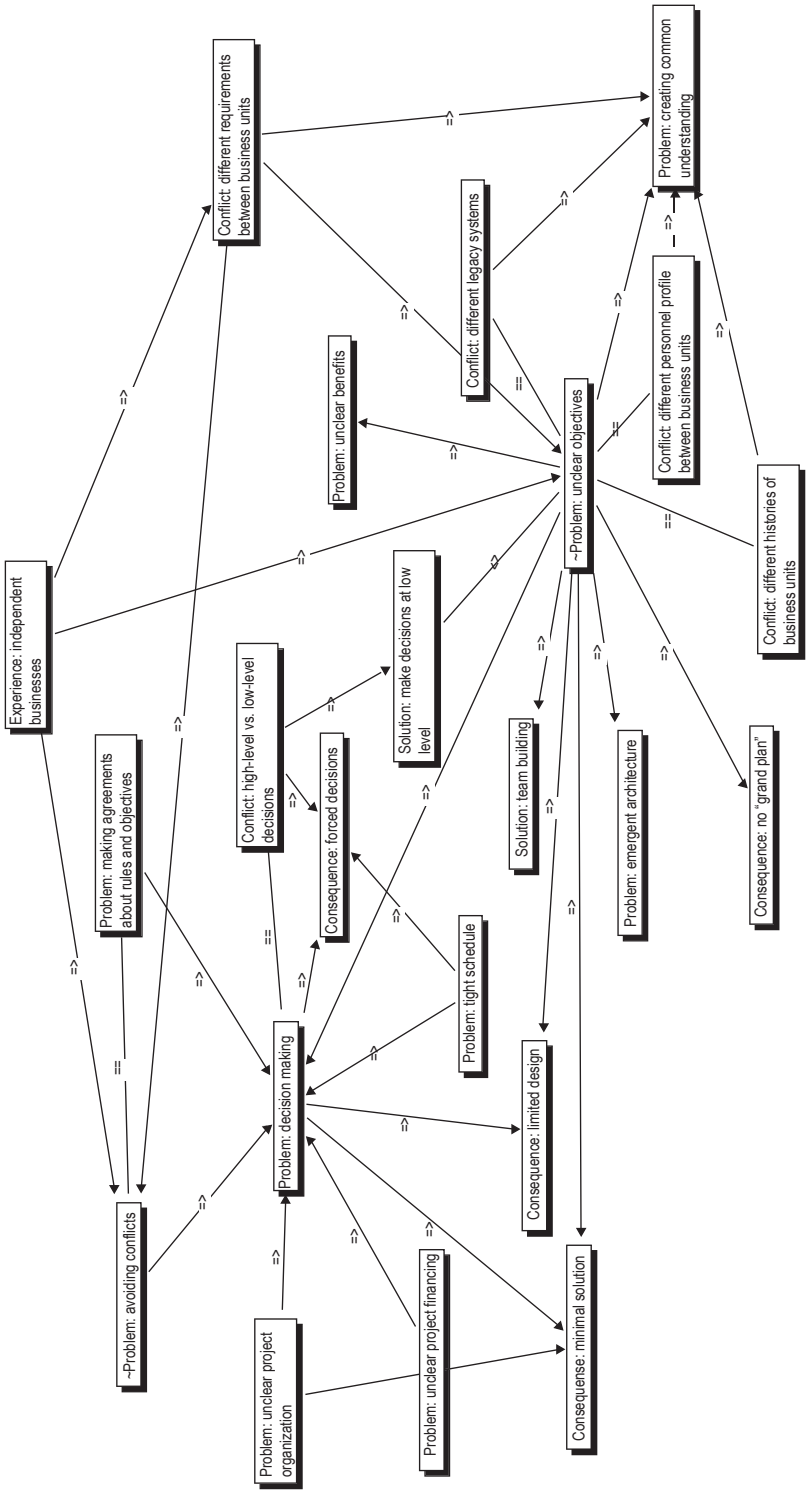
The axial coding started in parallel with the open coding and causal relationships between categories were recorded with Atlas.ti's semantic network capability. Figure 1 shows an example of such a network diagram. In the figure, the boxes represent categories, the arrows between them interpreted causalities, and the lines associations between categories. The number of categories and the number of identified relationships between the categories added up to 187 categories and 200 relationships, which created a problem of how to report such a multitude of categories and relationships. The solution was sought through abstracting out those categories that were rarely occurring in the data and interpreted as not so relevant regarding the research question. In addition, more attention was paid to those categories that occurred frequently in the data.

Inductively, we produced an explaining story to the events and forces under which the e-business development project had to work. The organization is facing market changes and changing the organization according to the changing markets. The objectives for the e-business development emerge from these changes and because the change is continuous and it brings all the time new requirements for the e-business system, the objectives are quite fluctuating. In addition, the history and legacy structures of the organization cause conflicts and problems in the development when combined with the need for change. These fluctuating objectives and emerging conflicts and problems brought certain consequences to the e-business architecture development in the organization. The formation and description of this explaining story can be considered as selective coding (Strauss & Corbin, 1990) and its details in the studied organization are explained in the next three sections.

The study has required extensive interpretation and exploration in the studied organization and therefore the main instruments of the research has been the researchers and their ability to interpret events and people's actions correctly. Robson (2002) lists three threats to validity in this kind of research, reactivity (the interference of the researcher's presence), researcher bias, and respondent bias, and strategies that reduce these threats. We have used these strategies in the following way:

- **Prolonged involvement:** Although this study lasted for one year, the research project altogether lasted for more than two years in the same organization and consisted of several phases and data collection rounds.
- **Triangulation:** The study has used data and observer triangulation as presented by Denzin (1978). To reduce the bias caused by researchers, we used *observer triangula-*

Figure 1. An example of a semantic network from axial coding



tion, because the data collection was done by two researchers. The bias caused by data was minimized using *data triangulation*, where different sources of data were used. Interviews were the primary data collection method, but we also received many kinds of project and company documents and architecture descriptions.

- **Peer debriefing and support:** The research has included regular meetings and discussions with involved research participants from several research institutions. In addition, preliminary results of research phases have been presented and discussed in conferences and workshops (Smolander, 2003; Smolander, Hoikka, Isokallio et al., 2002; Smolander & Päivärinta, 2002a, 2002b; Smolander, Rossi, & Purao, 2002, 2005).
- **Member checking:** The interpretation of the data has been confirmed by presenting the results to company participants in the research project.
- **Audit trail:** All interviews have been recorded and transcribed. The notes and memos of the study have been preserved and data coding and analysis results are available through the analysis tool used, ATLAS.ti.

CHANGES AND THEIR EFFECTS IN THE DEVELOPMENT CONTEXT

Starting Point: Changing Markets, Changing Organization

During the time of the data collection, there was a considerable change going on in the ICT market and the organization under study had undergone a deep change. A few years ago, the strategies emphasized growth and utilization of the possibilities in the stock market. This enforced independent business units inside the organization since the growth was easier to handle through independency. Each of the business units built independent e-commerce solutions and customer extranets, which resulted to a fragmentary set of e-commerce solutions to customers with own Internet sites, sales and billing systems, and Web-based customer support.

When the beliefs in the possibilities of ICT sector's continuing growth diminished, the organization had to change its strategies from growth to profitability and from stock market to customer orientation. With independent business units, there was no authority in the organization, which would see a customer as a whole. Instead, each business unit kept track of the customers only in the context of its independent business. To produce a unified customer interface a profound change to the way of building information systems and an integrated e-business solution was needed. This change would also require changes in business practices and organization. The organization should operate in a more integrated fashion and the barriers between independent units should be lowered.

The organization began to see technical e-business architecture as an enabler of change. The IS organizations in independent business units were obliged to cooperate and enforce commitment to the integration of information systems. This also emphasized the role of central information management, which had been in a minor role this far. Now, its roles would include the enforcement of information systems integration and enabling the unification of the sales channels and customer management for the planned e-business solution. At this point, the organization decided to establish a working group of systems architects from various parts of the organization. In the following section, we shall describe the context and the forces under which this group of architects were developing and designing the unified e-business architecture.

Conflicts, Problems and Varying Purposes

The context for e-business architecture development included many issues, which the working group for technical architecture development

had to face and be aware of. These included the market changes as described above, historical organizational inertia, fluctuating requirements and objectives, and conflicts and problems emerging from the market changes, inertia, and unclear objectives.

Historical Inertia

The organization's history with independent businesses and their diverging functions and objectives had both psychological and technical consequences causing slow progress and conflicts in the integrated e-business development. Each of the business units had legacy systems with incompatible information structures, technical architectures, and operating principles. It was not possible in practice to replace these systems with a uniform solution at once.

The historical inertia had effects also on the organization responsible for information management and information systems. Because of the independence, the organization had no clear central information management that could take responsibility of the e-business architecture development. Many of the conflicts and problems described later arose from this situation.

The Observed Objectives for the E-Business System

The fluctuating objectives, meanings, and requirements for the e-business architecture created another source of conflicts and problems. In a large organization with a high degree of independency, the conceptions among different business units and individuals about the purposes of an e-business solution vary considerably. Among the interviewees, we identified a large set of different purposes for the e-business system, which were then classified in five distinct classes:

- Creation of a unified electronic customer interface.
- Reduction of costs.
- Integration of information systems.
- Gaining business advantage.
- Implementing an organization change.

This list of observed purposes for the e-business system looks quite comprehensive and ambitious. Different interviewees emphasized the purposes differently and many saw that the only realistic objective was to implement a single sign-on procedure with a minimal level of customer information integration. The list anyhow shows the complicated and conflicting nature of objectives for the e-business system when it is developed for a large enterprise.

Emerging Conflicts and Problems

Changes in markets and organization, the history of the organization, and the complicated objectives for the e-business system put the architecture development group in a difficult situation. The group and its members were obliged to respond by some means and these responses shaped mitigated the role of deliberate design in the development process. In open coding, we identified in total 48 categories of conflicts and problems. This list was further combined to seven main categories, as follows:

- Varying requirements and unclear objectives
- Problems in the cooperation between technical and business people
- Conflict avoidance and problems in decision-making
- Problematic role of the central information management and its missing working practices
- Difficulties in creating common understanding about the architecture
- Difficulties in determining the level of integration
- Problems of implementing the integration

As described earlier, the purposes of the system were manifold and complicated and the requirements varied according to the business needs in the business units. The architects held this ambiguity of objectives and requirements as the biggest obstacle in the development. Those in the managerial level recognized the problem as well, but explained it as unavoidable in the

situation and expected that the first prototypes of the system will bring more clarity to the objectives. This resembles the chicken-egg problem: architects must know well the objectives to design the architecture, but the objectives are further clarified only after the first version of the architecture is built.

There were several mentions about the problems in the cooperation between technical and business people. Architects expected the business managers to explicate clear requirements and objectives for the system and its architecture. However, they considered the task impossible, because they thought that the business managers do not possess enough understanding about the possibilities of current technology. They felt that this leads to unrealistic objectives, which were manifested especially when considering the possibilities of legacy systems integration: people with business background had far more optimistic views than architects.

Conflict avoidance and problems in decision-making slowed the progress. Again, because of the history of independency, a central authority that could take care of the architectural decisions for the integrated e-business solution was missing. Because nobody took a full responsibility of the situation, this led to avoidance of conflicts and enforced the tendency towards compromises. A frequently occurring phrase among the architects included the term "lowest common denominator," which was usually noting to the compromised solution with a single sign-on procedure and a minimal level of customer information integration.

The role of the central information management was unclear and it was lacking the routine of large development efforts. The independency of businesses and the minor role of central information management had implications on the working practices. The architectural and development practices of the business units contained considerable differences implying that also common working practices needed to be established for the development process of the e-business system.

Even the understanding of the designed architecture and related technical solutions were difficult to communicate across the organization. Since the business units have had their own histories and produced their own legacy systems and information architectures, the interpretations on the situation and objectives diverged. This, combined with changing organization, unclear objectives, and missing common working practices, created difficulties in understanding and transferring architectural knowledge between the participants from different business units.

It was also difficult to determine the level of integration between the systems. The ownership of the information becomes an issue even in the most modest single sign-on e-business solution serving the whole organization. The question becomes, "who owns the customer information?" and relates to determining the integration level to the currently independent back-end legacy systems. The more ambitious integration, the more out-of-control the customer information (and possibly other information too) shifts from the business units.

In addition to determining the integration level, the actual implementation of integration proved to be problematic. Since the diverging legacy systems could not be replaced, they all had to be interfaced. Of the seven conflicts and problems occurring when creating e-business architecture, only the problem of implementing the integration was mainly a technical problem. The others were more related to the change in organization and practices that happen when developing an e-business system in a large organization with independent businesses. In the following, we shall look closer on what consequences these conflicts and problems cause for the architecture design and development process.

CONSEQUENCES: LIMITED DESIGNS AND MINIMAL SOLUTIONS

In the beginning of the project a unified architecture was seen as a panacea for solving the

problems of systems integration, streamlining the organization and unifying the customer interface. However, during the project it became clear that the aforementioned conflicts and problems would have some unfavorable consequences. While it was of paramount importance for the company to be able to streamline its systems and develop a more coherent architecture enabling the creation of an e-business system, the realities of legacy systems and the organization led to situation where it was best to seek satisfying, even minimal, solutions instead of optimal ones.

In the early phases of the project architecture was seen as general blueprints or roadmaps, largely drawn from scratch. Soon, however, the technical experts realized that evolutionary prototyping was the only possibility for progress in the architecture development. Because the schedule was tight, the objectives and requirements unclear and changing, and because the business units were rather independent, it was hard to achieve common understanding and commitment. With prototyping, it would be possible to clarify objectives and commit stakeholders by showing them visible results and benefits. This could be seen as "extreme" architecture design (Merisalo-Rantanen, Tuunanen, & Rossi, 2005). This could however lead to new problems. The technically oriented architects were specially worried that, combined with the quarter-based reporting system in the organization, evolutionary prototyping can easily produce quick-and-dirty and ad hoc solutions. We could classify the interviewees to those with positive attitudes towards prototyping and to those with negative or doubtful attitudes. In general, the project management believed positively that "somehow" the prototypes would transform to the final e-business solution, whereas technical architects presented more doubts and wanted to have explicit requirements and objective statements before committing to certain architectural solutions.

Prototyping and minimal solutions formed a vicious circle that made the development of robust and clear architectures nearly impossible by severely limiting the options available for the architecture developers. Existing legacy systems, the evolutionary approach, varying requirements, unclear objectives, difficulties in creating common understanding, and problems in decision making created a complex situation where textbook methods, description languages, and rational architecture design, as it is conceived in the literature, had no possibilities for immediate success. The degrees of freedom of design became limited. The system and its architecture could not be designed rationally as a whole, but rather one needed to accept the conditions and limitations caused by the factors above and to keep the day to day operations running while the new systems are continuously created through evolution.

The situation had also organizational consequences. We found clear hints of low-level networking and formation of shadow organizations as the result of unclear project organization and problems of decision-making and objective setting. As the organization and responsibilities change, new and perhaps inexperienced persons come into crucial official positions related to the e-business development. At the same time, the experienced architects and other key persons continued to stay in contact with each other. This unofficial shadow organization balanced the mismatch in skills and experience that might otherwise seriously impede the development.

The final consequence from all the above is, that in fact the e-business architecture becomes emergent: it is created gradually through compromises, constraints, and conflicts (c.f., Ciborra, 2000; Hanseth, Monteiro, & Hatling, 1996). The exact objectives and responsibilities will be resolved as the architecture emerges through evolutionary prototyping. Compared to the conventional view on software architecture design (Hofmeister et al., 1999a), most of the claimed benefits of rigorous architecture development seem to be lost. There is no "grand plan" since the work is proceeding in a day-to-day basis and the well defined responses and interfaces between systems do not necessarily emerge in a rationally planned way, but rather most duplicate functions are kept and there is

agreement only on a few items that become the "architecture."

DERIVED REQUIREMENTS FOR E-BUSINESS SYSTEMS DEVELOPMENT METHODOLOGY

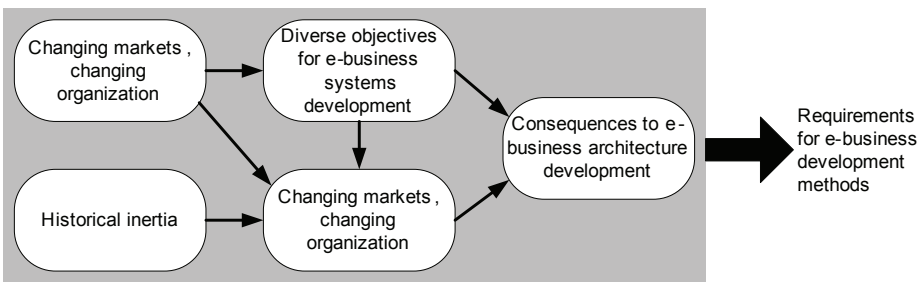
From the previous observations and explanations, we can derive a set of requirements that an e-business systems development methodology should meet. The grounded theory process resulted in an explanation model (Figure 2), from which a set of methodological requirements can be extracted. Changing markets and organization, historical inertia, and unclear objectives for the development produced a complex combination of conflicts and problems that brought various difficult consequences to the e-business development process. We analyzed the complex socio-technical situation and its consequences and reasoned the set of most pertinent methodological requirements. This was done by identifying and coding the methodological requirements in the interview transcripts and further combining them in 13 requirements as described below.

According to Lyytinen et al. a design methodology should conform to a set of key requirements (Lyytinen, Smolander, & Tahvanainen, 1989). It must embed several conceptual structures and description languages, and support several levels of abstraction at which the development process takes place. It should also cover the whole spectrum of activities in information systems development (ISD), include a prescribed model of activities to be carried out during the development process, include a model of the organizational form of the development (a set of human roles), and try to reuse existing descriptions and implementations. Tools for drawing, manipulating, and managing the descriptions should also support the methodology, in a balanced manner.

We can further elaborate this conception of ISD methodology by distinguishing between three separate contexts in ISD, namely the technical, language, and organization contexts (Lyytinen, 1987). The technical context is concerned with the technical components of the system (like hardware and software), language context forms the environment for linguistic communication, and the organization context provides the environment for systematic human interactions, including decision-making and operative control. An ISD methodology includes assumptions, models, languages, and tools related to these three contexts. In the following, we shall extract from the case the general requirements for e-business development methodology and classify them according to these contexts. The objective of this classification is to illustrate the nature and requirements of e-business architecture development in large organizations with several business areas and to highlight the areas with a weak methodical support.

Lyytinen commented already in 1987 that most development methodologies have too limited scope and they tend to concentrate on

Figure 2. Deriving the methodology requirements



technological issues late in the development lifecycle (Lyytinen, 1987). This limited scope omits most of the institutional and governance issues which seemed to be central for most stakeholders according to this study on architectural practice. One could argue that the organizational context is particularly relevant for e-business area, as most proponents of e-business emphasize the changes it brings about to work processes and organizations (Kalakota & Robinson, 2001).

The research into e-business architecture development is in a relatively immature stage. Previous literature has largely assumed that it solves technical issues for known problems (Taylor, McWilliam, Forsyth, & Wade, 2002). However, from the previous passages it has become obvious that methods for forming the problem statement and reaching a mutual agreement on what the architecture is in the end of the day are crucial. In this section, we take this as a starting point and observe the issues that rose in the described case starting from the inner, technical context and ending to the general organizational issues. This corresponds to Lyytinen's idea that the contexts are hierarchically ordered, because languages are presented by material carriers of technology context and language is needed for organized social action (Lyytinen, 1987). We identify e-architecture approaches in these areas and show how they propose solutions to the issues raised in our study.

In the following, we shall present the methodological requirements for each context. We also refer to the rows in Table 1 with the notation R1-R13.

Requirements from the Technology Context

Observed Requirements

The technical requirements of e-business development methods do not differ much from those of methods for traditional transaction-based information systems. E-business system development includes methodical requirements concerning e.g. distribution, error recovery, and networking, but those requirements can be met without a special "e-business support." A standard way to describe such technical solutions is of course required /R1/.

Integrated e-business architecture necessitates the integration of information systems in the organization and the rationalization of technology and development processes. Existing legacy systems will be integrated to the e-business functionality. This requires the selection of an integrative technology and the construction of development processes supporting the implementation of the integration. Because the integration is the basis and characteristic to e-business development, the development methodology should have specialized and usable techniques for describing information systems integration /R2/.

The key issue in the development of e-business systems is the keeping of the day-to-day operations running and at the same time implementing the integration between existing legacy systems and the new e-business functionality. This means that the nature of development is in many cases more analogous to a maintenance project than to a green-field development project. Current systems development methodologies and models of thought are mostly aimed at designing new systems instead of changing existing ones. This problem has been recognized before the advent of e-business, but it becomes more critical in the e-business development. From this we can derive a requirement that the development methodology for e-business systems should support evolutionary approaches to architectures and systems /R3/.

Existing Solutions

Most research on e-business systems development in general, and e-business architecture in particular, concentrates on this view. Much of the support that UML and RUP or their derivatives provide seems to concentrate on this area. Component aware methodologies, such as the Catalysis extension to UML, seem suitable for e-business. In addition, there are UML 2.0 extensions, such as SysML (Object Management Group, 2006), that provide better support

for technical architecture design. Bischler and Segev (Bichler et al., 1998) investigate the possibilities of component oriented approach for e-business. They take a technical viewpoint, and provide a useful listing of enabling technologies for e-business. An applicable standard in this area is the SysML extension to UML (Object Management Group, 2006). A work by Rossi & Schwabe (Rossi & Schwabe, 2000) uses patterns and frameworks as building blocks for e-business systems. This kind of approach could be particularly useful for a relatively well-specified domain, such as trade processes, which are assumed to be generic in nature. Baskerville & Pries-Heje see a relatively fixed architecture as a common ground, on top of which e-business systems can be built (Baskerville & Pries-Heje, 2001).

As mentioned earlier, in the e-business domain there are several layers of components available. The InterNCA architecture in (Lyytinen, Rose, & Welke, 1998) describes some of these and outlines needs for new breed of development methodologies, which would take into the account the particular problems of e-business systems development. Greunz & Stanoevska-Slabeva present an extension of UML, which can be used to realize systems on top of "media platform" architecture (Greunz & Stanoevska-Slabeva, 2002).

Requirements from the Language Context

The language context provides a means and an environment for linguistic communication which encompasses the use, nature, content, context and form of signs (Lyytinen, 1987). The methodology requirements coming from the language context deal with the ability of stakeholders to communicate successfully during the e-business architecture development process.

Observed Requirements

The chicken-egg problem between objectives and architecture becomes problematic in e-business development. To design a robust technical architecture, one must have clear objectives, and to select realistic objectives, one must understand the possibilities of the technical architecture. To overcome this problem, it is necessary to have a close cooperation between technical architects and those responsible of the business. This, however, induces a language problem. These groups often do not have a common language. To overcome the language problem, we need architecture description languages that business managers understand /R4/ and business descriptions that are explicit enough for technical people /R5/.

The problems of objectives and integration culminate on architecture design because the designs and prototypes related to technical architecture become the first concrete artifacts in the development showing implications of decisions to businesses and to the information management. Before architecture design, the plans and designs have been on the "PowerPoint presentation" level, showing ambiguous and general roadmaps and noble objectives. The more concrete the architecture becomes, the more various stakeholders become aware of the consequences, conflicts, and problems they will be facing. This leads to two distinct requirements for the development methodology: the methodology should take the development to a very concrete level (both politically and technically) very soon after the project initiation /R6/ and the architecture designs and descriptions (and their implications) should be approachable and intelligible by the various stakeholders participating the process /R7/.

Existing Solutions

As a description language, UML and its extensions offer a fairly strong support for engineering in the language context. Yet, there are very few articles describing these issues of having a common language in e-business area, but one could expect that methodologies used in other domains for participative processes and joint application development could be applied here (August, 1991). In this context, architecture serves as a language between the participants in the development process, enabling communication and making the consequences of the implementation

concrete to the participants. Using architecture as an enabler of communication between a diverse set of participants (including various levels of management and technical experts) requires informal and expressive approaches, which are practically non-existent in the field of software architecture research. This kind of conception of "architecture as language" can be associated with approaches that include rich and informal description techniques, like "rich pictures" in (Wood-Harper, 1985), the wall-charting technique (Saaren-Seppälä, 1988), and genre-based approaches (Päivärinta, Halttunen, & Tyrväinen, 2001).

Requirements from the Organization Context

Observed Requirements

These problems formed the largest bulk in our study. They included issues such as organizational inertia as well as environmental limitations, characteristics of a given business environment, codes of conduct in business, and regulatory and societal factors. These factors form together the 'ballpark' for an organization to act in relationship with its providers and customers.

The first organizational requirement comes from the overall conclusion of the case. The transition from heterogeneous e-commerce to integrated e-business is not only technically challenging. It is more a profound change to the organization. In fact, the primary challenge is in the change of the organization, not in the implementation of the technology. Therefore, e-business systems development methodology should support also the description of organizational change /R8/.

In this change of organization and implementation of technology, the role of central information management or some kind of central authority in the organization is crucial. The central authority should take care of the multitude of conflicts occurring when aiming at integration and coordinate the creation of objectives for the system. An e-business development methodology should enable the creation of a common vision /R9/, which can then be enforced by the central authority.

Evolution with modest but growing objectives may be the only way to develop integrated e-business systems. To foster commitment, some immediate benefits should be shown with the prototypes for each stakeholder. However, at the same time, the path to robust architecture should also be secured and enough time and resources must be given to technical architects. This very difficult and complex trade-off must be made in every e-business project /R10/.

The implementation of e-business integration deals not only with technical issues but also with difficult political ones. An organization shifting to integrated e-business must resolve issues concerning the internal ownership of information related for instance to customers, sales, contracts, and products. The ownership and responsibilities related to information must be decided and described during the development process. The development methodology should include descriptions for organizational responsibilities and ownership of information /R11/.

Identifying and agreeing about objectives became the most difficult problem in this case. Thus, to become valuable in practice, e-business development methodology should support not only the formation and recording of objectives but also measuring of success related to objectives /R12/.

The requirements directed to an e-business development organization are quite conflicting. On the other hand, the development requires a strong authority that can control the process through conflicts, and on the other hand, the formation of unofficial and shadow organization (peer-level networking) should be fostered to allow creative solutions and frictionless cooperation between businesses /R13/. This requirement is, however, not a new one when developing organizations.

Existing Solutions

From a more managerial and decision oriented view one could look at business- and strategy development methods, which aim at creation of

a common understanding and vision of business strategy. This view sees building of architecture as a common vision building effort rather than a system building effort. It could also be argued that e-business architecture building is quite similar to organizational change processes, especially the introduction of enterprise wide information systems, such as ERP. Koontz has argued for this by presenting e-business architecture development model, which is very generic (Koontz, 2000).

Organizational issues are largely neglected by the traditional systems development methodologies, but form important context and frame for the implementation of the e-business systems and architectures. The work on organizational change and observation of the power-play could be fruitful if applied to early stages of architecture development. However, they do merely observe the issues than provide solutions. Checkland's SSM methodology is one of the few general-purpose methodologies that identifies and models the "essence" of the organizational idea of the system and then proceeds to actual development of the system (Checkland & Scholes, 1990). It is clear from the observations in this case study that the explicit identification and framing of the problem to be solved, and then resolving the actual goals of the architecture forms the basis for architecture development.

Most studies thus far seem to assume that the development of e-architecture and infrastructure can be guided by the deliberate actions and decisions of management. However, as can be seen here the technological changes often evolve from designers' and users' experience with such technologies and are often unpredictable (Ciborra, 2000).The problem of loosing the original target while developing partial solutions and prototypes (e.g., see R10) could be helped by explicitly recognizing emergent and opportunistic possibilities created on the process.

Summary of Issues

The list above shows that most solutions and research this far, has concentrated on the technical level. Unfortunately, most of the problems seem to be non-technical in nature, they are rather more of the linguistic or organizational. E-business cuts across functional borders in organization and is built on a complex infrastructure of ERP and legacy systems and it shares many of the challenges and opportunities of these organizational technologies.

Table 2 summarizes these derived requirements for e-business development methodology. The requirements and their rationale are described in the text above. The 'Type' column places the requirement to the appropriate context or contexts (T: technology, L: language, O: organizational). The last column in the table ("Support in RUP employing UML") analyzes how unified modeling language (Object Management Group, 2005) and the Unified Process (Rational Software Corporation, 2001) support the e-business specific characteristics of the development process. This is important, because UML and RUP together form the current methodological basis for many software organizations. The column shows that the support is generally poor. The e-business specific requirements are not met by UML and RUP —only the standard technical issues are well covered. This conclusion calls for method development supporting better these e-business specific requirements.

In the technical context we noted that e-business development would benefit from method enhancements in IS integration and evolutionary development. However, the language and especially the organization context appeared to have more importance in the development. In the language context, there was an urgent need for more understandable and concrete architecture descriptions that could be used among many groups involved in the process, including technical and non-technical people. The organization context appeared as the most important target for research and practical methodical improvements. In that context, we could identify a multitude of issues requiring improvements, including better understanding and usable methods for the design and implementation of organization

Table 2. Summary of the requirements for e-business development methodology

	Requirement	Type	Rationale	Support in RUP employing UML
R1	Technical issues (like distribution, error recovery, and networking) must be described in a standard way.	T	These issues will occur as in all modern systems development	Good; this is what UML and RUP are for
R2	Specialized techniques for describing the information systems integration	T	IS integration is characteristic to e-business development	Poor; no specialized technique for the description of integration in standard UML. Some UML 2.0 extensions are however available.
R3	The development methodology should support evolutionary approaches to architectures and systems.	L/T	The change and maintenance of existing systems forms a major part of the e-business systems development	Moderate; UML and RUP are mainly targeted at the development of new systems
R4	Architectural description languages that business managers understand	L	To enable realistic objective selection, business managers must have some understanding on architecture	Poor; the descriptions necessitate too much technical skills and knowledge
R5	Business descriptions that are explicit enough for technical people	L	To understand the objectives, technical people must have understanding on business	Moderate; no description techniques showing overall aggregate view
R6	The methodology should take the development to a very concrete level (both politically and technically) soon after the project initiation	T/L/O	The more architecture becomes concrete, the more stakeholders become aware of the consequences, conflicts, and problems	Good (technically), none (politically)
R7	The architecture designs and descriptions (and their implications) should be approachable and intelligible by the various stakeholders participating the process	L/O	To enable wide understanding to the consequences of architectural selections (cf. R4).	Moderate; no relevant description technique besides Use Case diagrams
R8	Support for the description of organizational change	O	e-business involves deep changes to organization	Poor; some thoughts of "organization engineering" in RUP's Business Architecture
R9	Support for the description of a common vision	O	Resolve conflicts, build objectives	Poor; no common language for all stakeholders
R10	Both prototyping and careful architecture design needed	T	Gain commitment and resolve objectives through prototyping, aim at robust architecture	Moderate; iterative basis in RUP, but its implementation is difficult in practice

continued on following page

Table 2.continued

	Requirement	Type	Rationale	Support in RUP employing UML
R11	Methodology should contain descriptions for organizational responsibilities and ownership of information	L/O	The ownership of information becomes an issue when aiming at e-business integration	Poor; only general thoughts
R12	e-business development methodology should support the formation and recording of objectives and measuring of success related to objectives	L/O	Identifying and agreeing about objectives is one of the most difficult issues in e-business development	Poor; the objectives are mostly supposed to be given to the development project
R13	The development process should support organizationally both effective control structures and flexibility	O	Strong authority is needed to handle the conflicts and unofficial structures for creative solutions	Poor; development organization "design" in a general level

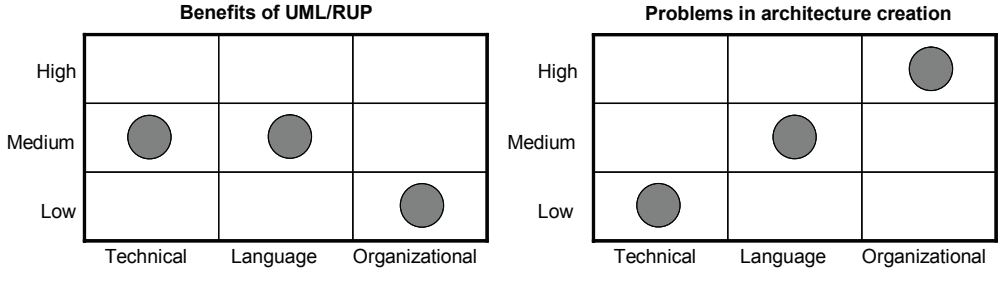
change, organizational vision, organizational ownership of information, and organizational responsibilities.

Figure 3 shows concisely our findings. When creating e-business or enterprise architecture, the major problems to be solved are organizational. This does not align with the support that UML and RUP provides, because they mostly concentrate on solving the problems in the language and technical contexts. It is the task of future research to provide improvements to this, but, as can be seen from Table 2, it might need quite radical extensions or changes to UML and RUP to be able to support effectively the formation of e-business architecture.

CONCLUSION

We have described a process where a large ICT company is building architecture for a comprehensive e-business system. From the case, we extracted 13 requirements for methodology supporting integrated e-business systems development and classified the requirements to technology, language, and organization contexts. We also compared the requirements to the support that UML and RUP offers and concluded that the e-business specific requirements are not met in UML and RUP. Successful e-business development requires alternative approaches that support better organization change, communication between stakeholders,

Figure 3. Support and requirements



systems integration, objective formation, and evolutionary development.

In our study, architecture manifested itself as a catalyst that makes business and organizational conflicts and problems concrete. When making decisions about architecture, the systems architects had to take into account the organizational situation in the company. At the same time the architecture starts shaping and changing the organization, thus forming a double mangle (e.g., Jones, 1998). The architects also realized that technical rationality is not enough for success in this kind of a situation. To succeed in e-business architecture development, one has to be aware of the political and organizational forces that are driving the development and its objectives. E-business architecture development can therefore be characterized as a process of seeking boundaries, finding sufficient consensus, and identifying commonalities across organizational borders. Most previous literature on architectural methods has neglected this and sought to develop description languages for describing the actual architectures for systems with clear problem statements, whereas we claim that it would be more important to seek tools that aid in building common understanding about the system and its architecture and tools for processing the emerging conflicts. Thus, we maintain that the field of architecture for e-business would benefit from tools that help to identify and process the emerging conflicts than tools that aid in developing a technically "perfect" and optimized solution. These tools could be used in early phases of development to augment UML and RUP based tools. Examples of such tools are group support systems and different participation facilitation systems. Thus we do not call for replacing UML, but rather adding tools that can be used to communicate with non-technical people about the architecture.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers of this paper for their valuable instructions and especially the reviewer that gave us the simple idea of Figure 3.

REFERENCES

August, J. H. (1991). *Joint application design: The group session approach to system design*. Englewood Cliffs, NJ: Yourdon Press.

Baskerville, R., & Pries-Heje, J. (2001, July 27-29). *Racing the e-bomb: How the internet is redefining information systems development methodology.* Proceedings of the IFIP TC8/WG8.2 Working Conference on Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspectice (pp. 49-68). Boise, Idaho.

Bichler, M., Segev, A., & Zhao, J. L. (1998). Component-based e-commerce: Assesment of current practices and future directions. *SIGMOD Record, 27*(4), 7-14.

Checkland, P. B., & Scholes, J. (1990). *Soft system methodology in action*. Chichester: John Wiley and Sons.

Ciborra, C. (2000). Drifting: From control to drift. In K. Braa, C. Sorensen & B. Dahlbom (Eds.), *Planet internet*. Lund: Studentlitteratur.

Conallen, J. (1999). Modeling web application architectures with UML. *Communications of the ACM, 42*(10), 63-70.

D'Souza, D. F., & Wills, A. C. (1998). *Objects, components, and frameworks with UML: The catalysis approach*: Addison-Wesley.

Dashofy, E. M., Van der Hoek, A., & Taylor, R. N. (2005). A comprehensive approach for the development of modular software architecture description languages. *ACM Transactions on Software Engineering and Methodology, 14*(2), 199-245.

Denzin, N. K. (1978). *The research act: A theoretical introduction to sociological methods*: McGraw-Hill.

Dori, D. (2001). Object-process methodology applied to modeling credit card transactions. *Journal of Database Management, 12*(1), 4.

Egyed, A., & Medvidovic, N. (1999, Oct). *Extending Architectural Representation in UML with View Integration.* Proceedings of the 2nd International Conference on the Unified Modelling Language (UML), (pp. 2-16). Fort Collins, CO.

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review, 14*(4), 532-550.

Fernández, W. D., Lehmann, H., & Underwood, A. (2002, June 6-8). *Rigour and relevance in studies of IS innovation: A grounded theory methodology approach.* Proceedings of the European Conference on Information Systems (ECIS) 2002, (pp. 110-119). Gdansk, Poland.

Garlan, D., & Kompanek, A. J. (2000). *Reconciling the needs of architectural description with object-modeling notations.* Proceedings of the Third International Conference on the Unified Modeling Language - UML 2000, (pp. 498-512). York, UK.

Glaser, B. (1978). *Theoretical sensitivity: Advances in the methodology of grounded theory*. Mill Valley: Sociology Press.

Glaser, B., & Strauss, A. L. (1967). *The discovery of grounded theory: Strategies for qualitative research*. Chigago: Aldine.

Greunz, M., & Stanoevska-Slabeva, K. (2002). *Modeling business media platforms.* 35th Annual Hawaii International Conference on System Sciences, Maui, HI.

Grinter, R. E. (1999). Systems architecture: Product designing and social engineering. *ACM SIGSOFT Software Engineering Notes, 24*(2), 11-18.

Hanseth, O., Monteiro, E., & Hatling, M. (1996). Developing information infrastructure: The tension between standardization and flexibility. *Science, Technology & Human Values, 21*(4), 407-426.

Hirschheim, R., & Klein, H. K. (1989). Four paradigms of information systems development. *Communications of the ACM, 32*(10), 1199-1216.

Hofmeister, C., Nord, R., & Soni, D. (1999a). *Applied software architecture*. Reading, MA: Addison-Wesley.

Hofmeister, C., Nord, R., & Soni, D. (1999b). *Describing software architecture with UML.* Proceedings of the First Working IFIP Conference on Software Architecture (WICSA1), (pp. 145-160). San Antonio, TX.

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process*. New York: Addison-Wesley.

Jones, M. (1998). *Information Systems and the Double Mangle: Steering a Course Between the Scylla of Embedded Structure and the Charybdis of Strong Symmetry.* IFIP WG8.2/8.6 Joint Working Conference, Helsinki, Finland.

Kalakota, R., & Robinson, M. (2001). *e-Business 2.0: Roadmap for Success*: Addison-Wesley.

Kazman, R., Klein, M., & Clements, P. (2000). *ATAM: Method for Architecture Evaluation* (Technical report No. CMU/SEI-2000-TR-004): Software Engineering Institute.

Kim, Y.-G., & Everest, G. C. (1994). Building an IS architecture: Collective wisdom from the field. *Information & Management, 26*(1), 1-11.

Koontz, C. (2000). Develop a solid e-commerce architecture. *e-Business Advisor*(January).

Kuntzmann, A., & Kruchten, P. (2003). The rational unified process—an enabler for higher process maturity. Retrieved April 19, 2007 from http://www-128.ibm.com/developerworks/rational/library/content/03July/0000/0579/Rational_CMM_WhitePaper.pdf.

Leist, S., & Zellner, G. (2006, April 23-27). *Evaluation of current architecture frameworks.* SAC'06, (pp. 1546-1553). Dijon, France.

Locke, K. (2001). *Grounded theory in management research*: SAGE Publications.

Lyytinen, K. (1987). A taxonomic perspective of information dystems fevelopment: Theoretical constructs and recommendations. In R. J. Boland, Jr. & R. A. Hirschheim (Eds.), *Critical issues in information systems research* (pp. 3-41): John Wiley & Sons.

Lyytinen, K., Rose, G., & Welke, R. (1998). The brave new world of development in the internetwork computing architecture (InterNCA): Or how distributed computing platforms will change systems development. *Information Systems Journal, 8*(3), 241-253.

Lyytinen, K., Smolander, K., & Tahvanainen, V.-P. (1989). *Modelling CASE environments in systems development.* Proceedings of CASE'89 the First Nordic Conference on Advanced Systems Engineering, Stockholm.

Martin, P. Y., & Turner, B. A. (1986). Grounded theory and organizational research. *The Journal of Applied Behavioral Science, 22*(2), 141-157.

Medvidovic, N., Egyed, A., & Rosenblum, D. S. (1999). *Round-trip software engineering using UML: From architecture to design and back.* Proceedings of the 2nd Workshop on Object-Oriented Reengineering (WOOR), Toulouse, France, Sept. 1999, 1-8.

Medvidovic, N., & Taylor, R. N. (2000). A classification and comparison framework for software architecture description languages. *IEEE Transactions on Software Engineering, 26*(1), 70-93.

Merisalo-Rantanen, H., Tuunanen, T., & Rossi, M. (2005). Is extreme programming just old wine in new bottles: A comparison of two cases. *Journal of Database Management, 16*(4), 41.

Monroe, R. T., Kompanek, A., Melton, R., & Garlan, D. (1997). Architectural styles, design patterns, and objects. *IEEE Software, 14*(1), 43-52.

Object Management Group. (1999). *UML Profile for Enterprise Distributed Object Computing: Request for Proposals (ad/99-03-10)*: OMG.

Object Management Group. (2005). *Unified modeling language: Superstructure version 2.0* (No. formal/05-07-04).

Object Management Group. (2006). *OMG SysML Specification (ptc/06-05-04)*.

Päivärinta, T., Halttunen, V., & Tyrväinen, P. (2001). A genre-based method for information system planning. In M. Rossi & K. Siau (Eds.), *Information modeling in the new millennium* (pp. 70-93). Hershey, PA: Idea Group.

Rational Software Corporation. (2001). Rational Unified Process [Online documentation, Version 2001A.04.00].

Robson, C. (2002). *Real world research, (2nd ed.)*. Blackwell Publishing.

Ross, J. W., Weill, P., & Robertson, D. C. (2006). *Enterprise architecture as strategy: Creating a foundation for business execution*: Harvard Business School Press.

Rossi, G., & Schwabe, D. (2000). Object-oriented web applications modeling. In M. Rossi & K. Siau (Eds.), *Information modelling in the next millennium*. Hershey: IDEA Group Publishing.

Rumpe, B., Schoenmakers, M., Radermacher, A., & Schürr, A. (1999). *UML + ROOM as a Standard ADL.* Fifth IEEE International Conference on Engineering of Complex Computer Systems, (pp. 43-53).

Saaren-Seppälä, K. (1988). *Wall chart technique: The use of wall charts for effective planning*. Helsinki: Kari Saaren-Seppälä Ky.

Sauer, C., Southon, G., & Dampney, C. N. G. (1997). *Fit, failure, and the house of horrors: Toward a configurational theory of IS project failure.* Proceedings of the eighteenth international conference on Information systems, (pp. 349-366). Atlanta, Georgia.

Shaw, M., & Garlan, D. (1996). *Software architecture: Perspectives on an emerging discipline*: Prentice Hall.

Siau, K. & Cao, Q. (2001). Unified modeling language (UML) — a complexity analysis. *Journal of Database Management, 12*(1), 26-34.

Siau, K., Erickson, J., & Lee, L. Y. (2005). Theoretical vs. practical complexity: The case of UML. *Journal of Database Management, 16*(3), 40-57.

Smolander, K. (2003, January 6-9,). *The birth of an e-business system architecture: Conflicts, compromises, and gaps in methods.* Hawaii International Conference on System Sciences (HICSS'36), Hilton Waikoloa Village, Big Island, Hawaii.

Smolander, K., Hoikka, K., Isokallio, J., Kataikko, M., & Mäkelä, T. (2002, April, 8-11). *What is included in software architecture? A case study in three software organizations.* Proceedings of 9th annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (pp. 131-138). (ECBS) 2002, Lund, Sweden.

Smolander, K., & Päivärinta, T. (2002a, May 27 - 31). *Describing and communicating software architecture in practice: Observations on stakeholders and rationale.* Proceedings of CAiSE'02 - The Fourteenth International Conference on Advanced Information Systems Engineering, (pp. 117-133). Toronto, Canada.

Smolander, K., & Päivärinta, T. (2002b, Aug 25-30). *Practical rationale for describing software architecture: Beyond programming-in-the-large.* Software Architecture: System Design, Development and Maintenance - IFIP 17th World Computer Congress - TC2 Stream / 3rd Working IEEE/IFIP Conference on Software Architecture (WICSA3), (pp. 113-126). Montréal, Québec, Canada.

Smolander, K., Rossi, M., & Purao, S. (2002, December 18). *Software architecture: Metaphors across contexts*. AIS Theory Development Workshop, Barcelona.

Smolander, K., Rossi, M., & Purao, S. (2005, May 26-28). *Going beyond the blueprint: Unraveling the complex reality of software architectures*. 13th European Conference on Information Systems: Information Systems in a Rapidly Changing Economy, Regensburg, Germany.

Sowa, J. F., & Zachman, J. A. (1992). Extending and formalizing the framework for information systems architecture. *IBM Systems Journal, 31*(3), 590-616.

Star, S. L., & Griesemer, J. R. (1989). Institutional cology, "translations" and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-39. *Social Studies of Science, 19*, 387-420.

Strauss, A. L., & Corbin, J. (1990). *Basics of qualitative research: Grounded theory procedures and applications*. Newbury Park, CA: Sage Publications.

Taylor, M. J., McWilliam, J., Forsyth, H., & Wade, S. (2002). Methodologies and website development: A survey of practice. *Information and Software Technology, 44*(6), 381-391.

Wood-Harper, T. (1985). Research methods in information systems: Using action research. In E. Mumford, R. A. Hirschheim, G. Fitzgerald & T. Wood-Harper (Eds.), *Research methods in information systems*. New York: North-Holland Publishers.

Kari Smolander is a professor of software engineering in the Department of Information Technology, Lappeenranta University of Technology, Finland. He has a PhD (2003) in computer science from Lappeenranta University of Technology and a Licentiate (1993) and Master (1988) degree from University of Jyväskylä, Finland. In addition to his long teaching experience, he has worked several years in industry and in 1990's he was the main architect in the development of MetaEdit CASE tool. He has published more than 50 refereed research papers in international journals and conferences. His current research interests include architectural aspects of systems development and organizational view of software development.

Matti Rossi is a professor of information systems at Helsinki School of Economics. He has worked as research fellow at Erasmus University Rotterdam and as a visiting assistant professor at Georgia State University, Atlanta. He received his PhD degree in business administration from the University of Jyväskylä in 1998. He is currently the European and African representative in the Association of Information Systems. He has been the principal investigator in several major research projects funded by the technological development center of Finland and Academy of Finland. His research papers have appeared in journals such as CACM*,* Journal of AIS*,* Information and Management*, and* Information Systems*, and over thirty of them have appeared in conferences such as ICIS, HICSS and CAiSE.*