Distributed Cognition in the Management of Design Requirements[1]

Sean Hansen
Kalle Lyytinen
Case Western Reserve University

Introduction

In the field study conducted as part of the initial *Design Requirements Workshop* (DRW), Hansen, Berente, & Lyytinen (2007) sought to explore the current state-of-the-art in requirements practice and identify areas for further research in the years ahead. In so doing, we uncovered a series of trends pushing design practice into new and little-explored avenues. One of the primary trends in requirements practice was the "increased distribution of requirements processes across functional, organizational, and geographic boundaries." Specifically, we found that within contemporary design environments it is difficult to isolate requirements activities in a single group or set of stakeholders. Rather, requirements come from a wide range of sources and the processes employed to generate and manage those requirements reflect equally diverse origins.

In the current position paper, we pursue a more focused discussion of the phenomenon of distributed requirements processes. We introduce a theoretical framework for understanding the mechanisms by which requirements processes are distributed socially, structurally, and temporally within systems development projects. This distributed requirements theory builds upon the theory of distributed cognition pioneered in the study of cognitive science (Hutchins, 1995, Hutchins and Klausen, 2000). To advance the development of the distributed requirements model, we start with a brief overview of the theory of distributed cognition. We then move to an articulation of the theoretical model itself.

The Theory of Distributed Cognition

Distributed cognition is a branch of cognitive science that emerged in the late 1980s and 1990s (Hutchins, 1995a; Norman, 1994; Salomon, 1993). The central feature of the theory is the rejection of the traditional assumption that cognitive processes, such as memory, decision making, and reasoning, are limited to the internal mental states of an individual. Rather, such cognitive processes are understood to be distributed across individuals engaged in collaborative tasks, between people and the artifacts that they employ, and even between individuals and features of the natural environment. While multiple perspectives exist on the distribution of cognitive processes, the *theory of distributed cognition* has largely been associated with the work of Edwin Hutchins and his colleagues, who were among the first to develop a well-articulated framing of the perspective within cognitive science (Hollan, Hutchins, & Kirsh, 2000; Hutchins, 1990, 1995a, 1995b; Hutchins & Klausen, 2000).

[1] The present theory development effort draws heavily upon the case studies provided to participants in the workshop: "The Case SIS Project: An Enterprise System in Higher Education" (hereafter, Case SIS) and "The Summit County Integrated Public Safety Initiative: Information Sharing in Law Enforcement" (hereafter, IPSI).

Dagstuhl Seminar Proceedings 08412
Perspectives Workshop: Science of Design : High-Impact Requirements for Software-Intensive Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1930

The development of the theory of distributed cognition was initially motivated by research on work teams engaged in complex time sensitive tasks with clear operational criteria for success. Early focal groups included navigation personnel on a naval aircraft carrier (Hutchins, 1995a) and airline cockpit crews (Hutchins & Klausen, 2000). Other research has applied the model to additional work contexts, including air traffic control towers (Halverson, 1995) and engineering work teams which is closest to typical RE tasks (Rogers, 1993). The salient observation from all of these settings is that information gathering and processing activities are not localized to individual members of a work team, but are distributed across all members and the artifacts that they employ. Furthermore, a significant portion of what might be characterized as the cognitive workload of a group is "shouldered" by the technical artifacts employed by group members.

Rising as it does from the field of cognitive science, distributed cognition adopts the principle metaphor of the domain – cognition as computation. Thus, intelligent behavior (i.e., cognition) is understood as an information-processing mechanism that results in appropriate action toward the achievement of specific goals – "the mind is a special kind of computer, and cognitive processes are the rule-governed manipulations of internal symbolic representations" (Van Gelder, 1995: 345). However, the theory of distributed cognition builds upon this conception from a new angle suggesting that an understanding of distributed cognition eliminates the boundary between internally-executed computation/cognition and computational mechanisms employed in the external world. This results in the formal distributed definition of cognition as "the propagation of representational state across representational media" (Hutchins, 1995a: 118).

This reframing of cognitive activity eliminates the arbitrary boundary traditionally applied to cognitive processes – i.e., the human skull. In so doing, the theory of distributed cognition lends itself to at least three profound assertions regarding the processes of thought in action (Hutchins, 2000): the distribution of thought among members of social groups, cognition employing both internal and external structures, and cognitive distribution over time. Cognitive processes are socially distributed when each member of task-oriented team plays a specific role in the collection and processing of information and the initiation of action by the group. Structural distribution (which Hutchins refers to as "the interplay of internal and external structure") occurs in so far as individuals and groups integrate physical elements of the environment as part of their thought processes. This form of distribution reflects what Clark (1997) refers to as *scaffolding* – the use of physical artifacts to enhance internal cognitive processes. Finally, cognitive processes are temporally distributed in that the outcomes of earlier actions can influence the cognitive activities employed in later tasks. A thorough analysis of contemporary requirements practices in systems development efforts reveals that all three forms of distributed cognition are readily apparent.

Cognitive Distribution in Contemporary Requirements Practice

As noted above, the theory of distributed cognition centers on an understanding of cognition as the propagation of representational state across representational media. Employing this understanding, we see a wide and varied distribution of cognitive processes within prevailing requirements practice. To support our understanding of the nature of these distributed cognitive processes within the domain of requirements engineering, we have developed theoretical model based on detailed observation of the day-to-day efforts of practicing systems development teams. This distributed requirements theory is graphically depicted in Figure 1. We will briefly discuss the mechanisms of distribution reflected in the theoretical framework.

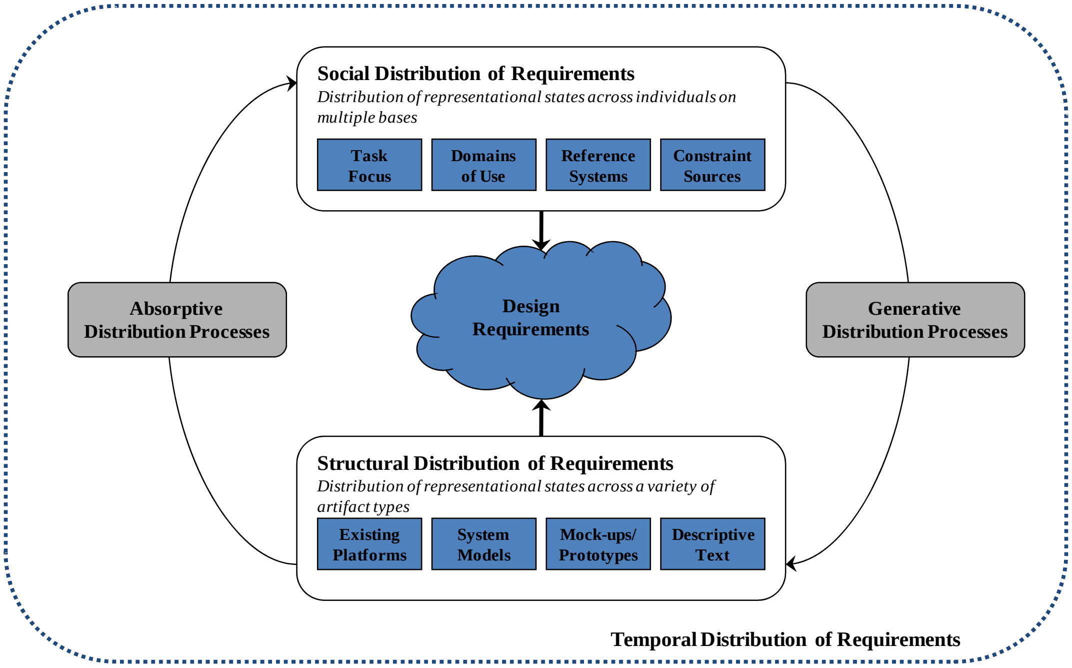

Figure 1. *Process Model of Distributed Requirements Practice*

Social Distribution of Requirements

The social distribution of cognitive activity has perhaps the most obvious application to the study of distributed requirements processes. Nearly all systems development efforts are executed through a team structure (Guinan, Cooprider, & Faraj, 1998), and one of the most essential characteristics of systems development teams is the diversity of knowledge that they require (Curtis, Krasner, & Iscoe, 1988; Levina & Vaast, 2005; Walz, Elam, & Curtis, 1993). To address the complex challenges set before them, design teams bring together individuals from a wide variety of technical and functional domains. Within systems development projects, the social distribution of requirements processes can be seen in several distinct mechanisms by which requirements efforts are segmented.

Task focus. On nearly any project of significant size, systems development teams employ a division of labor around the tasks pursued. Common structures for such task-based separation of labor include focused attention to the facets of IT architecture, solution design, technical development, and project management or coordination. Each of these areas is supported by dedicated personnel with a well-constrained primary focus. To maintain a common vision of the broader design/development effort, the various task-focused sub-teams or individuals are brought together through regular mechanisms for personal interaction and exchange of perspectives, such as project meetings and walkthroughs.

Domains of use. While the primary focus of the present theory is on the cognitive distribution of requirements processes, this phenomenon is integrally tied to other prevailing trends in contemporary systems development discussion by Hansen et al. (2007), including the interdependent complexity[2] of systems and an emphasis on integration over raw development in most projects. In many current development efforts, we see that project personnel are segmented based on the envisioned domains of use. In many cases, members of a development team or selected representatives of the relevant "lines

[2] By *interdependent complexity*, we mean to capture the inherent complexity involved in integrating interdependent systems and components.

of business" are tasked with eliciting, specifying, and validating requirements for a given application domain. These individuals are identified as the subject matter experts (SMEs) within the given domain. In one of the cases studied in the development of the present theoretical framework (see the case study summary for *The Case SIS Project*, available to workshop participants), examples of the domains of use on university-based development effort include admissions, registration, and faculty domains, as well as those associated with specific schools within the university.

Reference systems. On projects where multiple legacy systems must be replaced or integrated, development personnel may also be segmented based on the reference systems involved. That is, dedicated individuals are assigned to oversee the requirements analyses associated with distinct legacy platforms as well as staff who are focused on the functionality of relevant commercial-off-the-shelf (COTS) platforms. This again reflects the trend toward integration and interdependent complexity in contemporary development projects. This form of project team segmentation is observable in both of the case studies provided to the seminar participants. For example, on the Case SIS Project, dedicated personnel were assigned to capture and communicate the integration requirements for each of the primary legacy systems intended for integration with the PeopleSoft platform.

Constraint sources. Finally, a significant number of constraints sources provide a framework for division of labor. In the parlance of requirements engineering researchers, these reflect the sources of significant non-functional requirements. Many ISD projects today take place within organizations confronting a multitude of regulatory or compliance-oriented contexts. Development team members may be assigned to address the requirements impacts of these various compliance environments. Consider the U.S. financial services market (so hotly debated in recent weeks): Despite appearances, organizations within this marketplace are subject to the oversight of several federal-level regulatory agencies, and the complexity of the compliance landscape demands dedicated attention to each of the distinct regulatory domains. For example, in the IPSI case, dedicated project team members were responsible for integrating the governmental compliance requirements associated with arrest and prosecution, while others were focused on compliance with the Global Justice XML Data Model (GJXDM).

Structural Distribution of Requirements

The distribution of cognitive activity through the use of external structure is perhaps more prevalent in the development and use of software-intensive systems than in any other facet of contemporary business. Certainly, such structural distribution can be clearly discerned in prevailing requirements practice. Indeed, the development of formal models that has predominated in traditional requirements research can be understood as a mechanism for creating external structures that will support subsequent design efforts. Specific forms of structural distribution that we have identified in requirements efforts include the following:

Existing platforms. While the use of external structure is clearly apparent in requirements activities themselves, existing artifacts also serve as a significant *source* of design requirements. Legacy systems and enterprise architectures have become a critical jumping-off point for today's design requirements efforts – setting the initial conditions which both enable and constrain a design process (Zachman, 1997, Antón and Potts, 1998, Hansen et al., 2007). In addition, COTS platforms are brought into project efforts with essentially self-contained requirements sets – the products of the iterative requirements processes engaged by the vendor in question. Thus, both legacy and novel platform element embody a range of requirements that must be aligned for the specific demands of a focal development effort.

System models. Perhaps no single subject within requirements research has received more attention than that of modeling (van Lamsweerde 2000). Some even argue that model development lies at the very core of the entire requirements undertaking (Borgida et al. 1985). If so, perhaps the criticality of requirements modeling lies in its creation of a powerful external cognitive structure. Many of the argued advantages of formal models (e.g., enabling analysis to identify unstated requirements, predict behavior, determine inconsistencies between requirements, and check for accuracy) derive their value from the degree to which they enhance internal human cognitive functioning. In a very real sense, models are the central scaffolding of requirements practice.

Mock-ups/Prototypes. In recent years, the systems development field has seen a widespread adoption of CASE tools to support both requirements capture and traditional software design (Vessey and Sravanapudi, 1995, Kruchten, 2003). These platforms greatly enhance the development of mock-ups or more elaborate prototypes of various software modules. Prototypes are a clear form of structural distribution of cognitive requirements activity. They augment the cognitive processes of project members in the same way as systems models in that they fundamentally alter the internal cognitive processes applied by the human participants – changing the representational state of a problem such that elements of the solution become obvious (Simon, 1996).

Descriptive text. Finally, the most common mechanism for the creation of external structure in requirements processes, is natural language requirements documentation. Despite efforts by systems development practitioners and researchers to move requirements toward more formal and systematic specification, natural language remains a preferred mechanism for the communication and capture of requirements among system stakeholders (i.e., users; Hsia et al., 1993). Importantly, natural language still supports the cognitive functioning of a project team by providing a common basis for communications between the stakeholders and designers, as well as a great deal of information about application domains. In this way, natural language requirements descriptions act as a critical design boundary object – enabling the transfer of design knowledge and the alignment of interests across social worlds (Bergman et al. 2007; Star & Griesemer 1989).

Temporal Distribution

The third fundamental form that cognitive distribution of requirements processes takes is that of temporal distribution. This again reflects the idea that outcomes of earlier actions influence the cognitive processes enacted in later efforts. For example, upon undertaking a new line of research, a physicist does test the fundamental physical properties of a material, because he or she relies on information collected in existing texts – the cognitive tasks facing the physicist begin where earlier researchers left off. Similar illustrations of the distribution of cognition over time can be seen in almost any cultural context, where rules and heuristics have been developed through generations of cognitive activity.

Interestingly, Hutchins (2000) characterizes temporal distribution as a facet distinct from social and structural facets. However, in the present theory development effort, we perceive temporal distribution to be an outgrowth of (and essentially reliant upon) social and structural forms. Thus, distribution of cognition over time is possible because of the “memory function” provided by the social distribution (e.g., personal/team experience and expertise), structural distribution (e.g., existing systems or earlier project artifacts), and the interplay of the two.

Building upon this insight, temporal distribution of requirements-related cognition can be seen in nearly all systems development efforts. Contemporary requirements practices draw heavily upon the requirements and development artifacts from earlier design projects. For example, the creation of formal information architectures (e.g., enterprise and product architectures) is pursued as a mechanism for ensuring consistency across multiple design initiatives (Hansen et al., 2007). In addition, an extensive literature on requirements reuse provides multiple approaches to improving the distribution of requirements over time (Sommerville and Sawyer, 1997, Lam et al., 1997, Cybulski, 1998, Nuseibeh and Easterbrook, 2000). In the Case SIS Project referenced earlier, temporal distribution of requirements processes can be seen in the several examples, including: the intensive use of external consultants with experience implementing the PeopleSoft platform at other universities and the reliance upon the higher education user group (HEUG) forum for insights from earlier Oracle/PeopleSoft initiatives.

Movement between Representational States

Having discussed the basic social and structural representational states observed in contemporary requirements processes, the question remaining is how the propagation of these states is achieved. In the present framework, we identify two fundamental types of movement between representational states in the model: Generative and absorptive distribution process. We discuss each of these in turn.

Generative distribution processes. The generative distribution processes are those by which social actors engaged on the project (be they project team members or other stakeholders) develop new representational states through the creation of external artifacts. Within this broad category of processes, distinct forms can be identified, including: development, modification, and recombination. This externalization can be executed either individually or collectively. Perhaps, the simplest form of the generative process is the creation of new requirements artifact (e.g., business requirements document) by a single member of the project team. This may entail one or more of the structural forms of distribution discussed above (e.g., descriptive text, models). The critical point is that, in developing the artifact, the individual generates a new representational state for use by the broader cognitive system. Similarly, the modification of an existing artifact in a development team walkthrough, for example, results in the generation of a new representational state.[3] Finally, recombination reflects the development of novel states (again, artifacts) through the reuse of elements from earlier artifacts. Such processes are especially critical in the pursuit of requirements reuse. Thus, what we have labeled the generative distribution processes provide a mechanism for the movement from socially distributed cognition to structurally distributed cognition. In so doing, they largely encapsulate several facets of traditional requirements research including the umbrella category of "specification." Importantly, the stream does not flow in only one direction.

Absorptive distribution processes. The absorptive distribution processes are those by which existing artifacts (i.e., external representational states) are employed by the social actors on a project to propagate novel social representational state. Here again, multiple process can be identified, including discovery, derivation, and collaborative interpretation. *Discovery* has a long-standing role in requirements processes, being frequently used synonymously with the term *requirements elicitation*. However, in the current context, we refer to discovery as the process by which project stakeholders identify and integrate artifacts that are relevant for a give project effort. Derivation reflects the use of existing artifacts not for direct integration with the relevant project effort but to guide the development social representational states (e.g., project teams structures) or the generation of novel structural states (e.g., as a guideline for project-specific requirements documents). An example of derivation drawn from one of the cases provided to workshop participants (see, "The Summit County Integrated Public Safety

[3] This example also illustrates the dynamic interplay of generative and absorptive distribution processes.

Initiative") is the use of a request for proposals (RFP) document from an earlier project effort to support the determination of business need in the focal project. As this example illustrates, derivation is the absorptive counterpoint for modification. Finally, the social actors involved in requirements processes frequently engage in collective interpretation of existing artifacts, such as models or prototypes. This process of collective interpretation and discussion represents a mechanism by which the external representational state is employed to create understandings of the requirements through the socially distributed states of the system.

Through these various generative and absorptive distributions processes, the iterative movement between social and structural representational states is achieved. We had said above that the temporal distribution of requirements efforts is driven by the interplay of the social and structural forms of distribution. It is important to note that this temporal distribution is accomplished through the adoption of generative and absorptive processes.

Design Requirements

The final element of the theoretical model that warrants discussion is perhaps the most critical of all in that it is the objective of the entire cognitive system – the design requirements set. From the perspective of the theory of distributed cognition, objectives are the primary determinant of the unit of analysis (i.e., the composition of relevant cognitive system). For the present discussion, we shall assume that the objective of any requirements process is the development and maintenance of a set of design requirements. However, in a distributed cognitive model, this set is not merely reflected in the documentation (e.g., a specification document) that results from the requirements processes, but rather it is the collected understanding of the requirements for the system, as embodied both in the social actors and variety of artifacts that inform the design process. As such, it is important to note that the contours of the design requirements set for any given project are never entirely set. They are always subject to revision and reinterpretation. Furthermore, they are understood to be constantly evolving.

Implications

The theory of requirements as a distributed cognitive process opens up a range of insights into contemporary requirements practice. In addition to focusing the attention of researchers on the myriad ways in which requirements are distributed socially, structurally, and temporally, the theory suggests ways in which requirements practice can be enhanced. First, the theory helps us to move beyond traditional views of requirements as a centrally controlled and independent process. While most systems development projects still have a project team assigned to aggregate and management the design requirements set, it is increasingly apparent that the processes and focus of even these actors is far from uniform. Accordingly, the theory suggests that methods for improving requirements outcomes could be fruitfully reoriented toward the mechanisms by which interdependent individuals and teams (including all of the stakeholders on a given project) share or exchange ideas to propagate representations of the requirements.

Secondly, the theory enables use to see the interplay of individuals and artifacts as components of a broader design system. Rather than viewing artifacts as simple tools employed by social actors, we understand them as elements that bear a significant portion of the cognitive workload of the system. The artifacts fundamentally change the internal cognitive processes demanded of individuals participating on a project. Accordingly, one must consider all of the elements of the system (both social and structural) to understand the process of "thought" that requirements efforts entail. This leads us to questions about improving design environments. From the perspective of distributed requirement theory, we must explore the ways in which the propagation of representational states is achieved. How can the generative and absorptive processes of the system be enhanced?

Third, the theory provides us with a different perspective for organizing, understanding, and predicting the outcomes of requirements processes. Do particular forms of distribution support more effective requirements outcomes? Is there an optimal composition of the cognitive system – or does the desired composition vary based on the types of design efforts undertaken?

Finally, the framework gives a mechanism for integrating several of the emergent trends in requirement practice discussed during the first Design Requirement Workshop (Hansen et al. 2007). The growing focus on integration vs. traditional development, the centrality of architecture, the emphasis on business processes, the widespread preference for COTS components, the layering of requirements, and the fluidity of requirements and design can all be understood as measures designed to improve the functioning of the distributed cognitive processes of requirements and design. From this perspective, we can argue that practicing design professionals are pursuing methods that support more effective generation and absorption of various elements of the design vision.

Conclusion

In this position statement, we have outlined a new theoretical framework of the distribution of design requirements processes. In this theory, we characterize requirements efforts as distributed cognitive processes, employing social and structural elements in the pursuit of a unified vision of the requirements for a system design and development initiative. A wide array of questions remain to be answered: What implications does this model of requirements processes have for enhancing the practice of requirements management? What measures might be taken to improve the generative and absorptive potential of systems development project teams? What avenues of research might be opened up by the application of this model in the requirements domain? We are hopeful that these and more questions will be addressed during the workshop.

References

Antón, A. & Potts, C. 1998. The use of goals to surface requirements for evolving systems. Proceedings of the 20th international Conference on Software Engineering: 157-166.

Bergman, M., Lyytinen, K., and Mark, G. (2007). Boundary Object in Design: An Ecological View of Design. Journal of the Association for Information Systems, 8(11): 546-568.

Borgida, A., Greenspan, S., & Mylopoulos, J. (1985). Knowledge Representation as the Basis for Requirements Specifications. IEEE Computer, 18**:** 82-91

Clark, A. 1997. Being There. Cambridge, MA: The MIT Press.

Curtis, B., Krasner, H., & Iscoe, N. 1988. A field study of the software design process for large systems. Communications of the ACM, 31(11): 1268-1287.

Cybulski, J. 1998. Patterns in software requirements reuse. Paper presented at the Third Australian Conference on Requirements Engineering (ACRE'98), Geelong, Australia.

Guinan, P. J., Cooprider, J. G., & Faraj, S. 1998. Enabling Software Development Team Performance During Requirements Definition: A Behavioral Versus Technical Approach. Information Systems Research, 9(2): 101-125.

Halverson, C. A. 1995. Inside the Cognitive Workplace: New Technology and Air Traffic Control, Unpublished Dissertation, University of California-Sand Diego. Sand Diego, CA.

Hansen, S., Berente, N., & Lyytinen, K. 2007. Requirements in the 21st Century: Current Practice and Emerging Trends. Design Requirements Workshop, Cleveland, Ohio, USA.

Hansen, S. & Lyytinen, K. (2008). The Case SIS Project: An Enterprise System in Higher Education. Perspectives Workshop – Science of Design: High-Impact Requirements for Software-Intensive Systems, Schloss Dagtuhl, Germany, October 8-12.

Hansen, S. & Lyytinen, K. (2008). The Summit County Integrated Public Safety Initiative: Information Sharing in Law Enforcement. Perspectives Workshop – Science of Design: High-Impact Requirements for Software-Intensive Systems, Schloss Dagtuhl, Germany, October 8-12.

Hollan, J., Hutchins, E., & Kirsh, D. 2000. Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research. ACM Transactions on Computer-Human Interaction, 7(2): 174-196.

Hsia, P., Davis, A.M., & Kung, D.C. 1993. Status Report: Requirements Engineering. IEEE Software, 10**,** 75-79.

Hutchins, E. 1990. The technology of team navigation. In J. Galegher & R. Kraut & C. Egido (Eds.), Intellectual teamwork: social and technical bases of collaborative work. Hillsdale, NJ: Lawrence Erlbaum Assoc.

Hutchins, E. 1995a. Cognition in the Wild. Cambridge, MA: MIT Press.

Hutchins, E. 1995b. How a Cockpit Remembers Its Speed. Cognitive Science, 19: 265-288.

Hutchins, E. & Klausen, T. 1996. Distributed Cognition in an Airline Cockpit. In Y. Engestrom & D. Middleton (Eds.), Cognition and Communication at Work: 15-34. New York: Cambridge University Press.

Hutchins, E. 2000. Distributed Cognition, International Encyclopedia of the Social & Behavioral Sciences: Elsevier, Ltd.

Kruchten, P. 2003. The Rational Unified Process: An Introduction (Third Edition ed.). Boston, MA: Pearson Education.

Lam, W., McDermid, T., & Vickers, A. 1997. Ten steps towards systematic requirements reuse. Paper presented at the Third IEEE International Symposium on Requirements Engineering, Annapolis, MD, USA.

Levina, N. & Vaast, E. 2005. The Emergence of Boundary Spanning Competence in Practice: Implications for Implementation and Use of Information Systems. MIS Quarterly, 29(2): 335-363.

Norman, D. 1994. Things That Make Us Smart: Defending Human Attributes in the Age of the Machine. Cambridge, MA: Perseus Books.

Nuseibeh, B. & Easterbrook, S. 2000. Requirements engineering: a roadmap. Proceedings of the Conference on the future of Software Engineering: 35-46.

Rogers, Y. 1993. Coordinating Computer-Mediated Work. Computer Supported Cooperative Work, 1(4): 295-315.

Salomon, G. 1993. Distributed Cognitions: Psychological and Educational Considerations: Cambridge University Press.

Sommerville, I. & Sawyer, P. 1997. Requirements Engineering: A Good Practice Guide: John Wiley & Sons, Inc. New York, NY, USA.

Star, S. L., & Griesemer, J. R. (1989). Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39, Social Studies of Science, 19: 387-420.

Van Gelder, T. 1995. What Might Cognition Be, If Not Computation? The Journal of Philosophy, 92(7): 345-381.

van Lamsweerde, A. (2000). Requirements Engineering in the Year 00: A Research Perspective. Proceedings of the 22nd International Conference on Software Engineering, 5-19

Vessey, I. & Sravanapudi, A. P. 1995. CASE tools as collaborative support technologies. Communications of the ACM, 38(1): 83-95.

Walz, D. B., Elam, J. J., & Curtis, B. 1993. Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration. Communications of the ACM, 36(10): 63-77.

Zachman, J. 1997. Enterprise Architecture: The Issue of the Century. Database Programming and Design, 10(3): 44-53.