Distributed Cognition in the Management of Design Requirements

Sean Hansen
Case Western Reserve University, hansen@case.edu

Kalle Lyytinen
Case Western Reserve University, kalle@case.edu

Follow this and additional works at: http://aisel.aisnet.org/amcis2009

Recommended Citation
http://aisel.aisnet.org/amcis2009/266

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Distributed Cognition in the Management of Design Requirements

Sean Hansen  
Case Western Reserve University  
hansen@case.edu

Kalle Lyytinen  
Case Western Reserve University  
kalle@case.edu

ABSTRACT
Requirements processes in contemporary information systems development efforts have become increasingly distributed across organizational and temporal boundaries. This trend has created a range of challenges and opportunities for IT design professionals. In the present study, we combine the insights from a multiple-case study research effort and the framework provided by the theory of distributed cognition to develop a model of distributed requirements practice. The model reveals that systems development projects reflect social, structural, and temporal forms of distributed cognition as they pursue effective design outcomes.

Keywords
Requirements processes, distributed cognition, case study methodology.

INTRODUCTION
Requirements processes have remained among the most profound sources of difficulty in the design and development of information systems. Since the Standish Group first published its survey of information systems success and (more notably) failure (1995), researchers have been quick to note that the three leading sources of project distress – i.e., lack of user input, incomplete requirements, and changing specifications – are directly related to the creation and management of a project’s design requirements (Aurum and Wohlin, 2005; Leffingwell and Widrig, 1999; Crowston and Kammerer, 1998; Hickey and Davis, 2003; van Lamsweerde, 2000). Despite decades of research on the subject, the “requirements mess” continues to plague information technology professionals as they struggle to design artifacts in accordance with the needs and desires of system stakeholders (Lindquist, 2005).

Complicating this pursuit is the fact that the way in which requirements are discovered, articulated, validated, and managed continues to change, and the requirements research community has struggled to keep pace with developments in requirements practice (Siddiqi and Shekaran, 1996; Kaindl et al., 2002; Berry and Lawrence, 1998; Jarke and Pohl, 1994). With the rise of development outsourcing, a shift in emphasis toward commercial off-the-shelf (COTS) applications, and the widespread use of consultants and third-party systems integrators, the requirements processes for contemporary IS design initiatives are increasingly distributed across a wide range of individuals, teams, and organizations. Some researchers have referred to this development as the emergence of distributed requirements processes (Hansen et al., 2009). The observation of such distributed requirements phenomena presents a stark contrast to the traditional focus on requirements efforts as orchestrated by a single systems analyst with respect to the design of a standalone artifact (Jarke and Pohl, 1994).

While requirements researchers have begun to explore the impact of geographic distribution among members of a design team (Damian et al., 2003; Jarke and Pohl, 1994; Grünbacher and Braunsberger, 2003; Lloyd et al., 2002), little empirical work has been done to identify the ways in which requirements processes are distributed organizationally and over time. In the present study, we offer a more thorough exploration of the phenomenon of distributed requirements processes and a discussion of the challenges and potential benefits that such distributed processes engender. Specifically, we focus on the following research questions:

- What are the distinct ways in which requirements processes are distributed across organizational and temporal boundaries?
Distributed Cognition in the Management of Design Requirements

How do practicing designers manage the aggregation of requirements from heterogeneous sources with respect to promoting attention of specific user needs, ensuring consistency across requirements, and eliminating redundancy and waste in requirements efforts?

To explore and organize the various facets of distributed requirements processes, we apply the theoretical framework offered by the theory of distributed cognition (Hutchins, 1995a; Hutchins and Klausen, 2000). Drawn from the field of cognitive science, distributed cognition posits that cognition is not bounded by the skull of an individual thinker, but is distributed socially, technically, and temporally across individuals and artifacts within the external environment. Accordingly, distributed cognition offers a fruitful lens for the analysis of distributed requirements processes encountered in information systems development (ISD) efforts. 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.

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).

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.

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). 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 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 reflect social distribution 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 integrate physical elements of the environment as part of their thought processes. Finally, cognitive processes are temporally distributed in that the outcomes of earlier actions can influence the cognitive activities employed in later tasks. An analysis of contemporary requirements practices in systems development efforts reveals that all three forms of distributed cognition are readily apparent.
RESEARCH METHODOLOGY

Building upon the framework provided by the theory of distributed cognition, we have pursued an exploratory study of distributed cognitive processes as they are reflected in the day-to-day activities of practicing design teams. The distribution of requirements processes is a phenomenon that is emerging in real-world design environments that are not subject to straightforward manipulation. Accordingly, a case study approach is warranted to provide a rich exploration of the practical activities of systems designers (Eisenhardt, 1989; Yin, 2003). Indeed, for this very reason, the case study has been a favored approach when empirical work has been conducted within the broader domain of requirements engineering, often in an effort to validate a prescribed technique (Gotel and Finkelstein, 1997; Zave, 1997; Ramesh and Jarke, 2001).

For our data collection effort, we have employed a multiple-case study design. The unit of analysis for the case study effort is an individual design project. The case inquiry focused on requirements-related activities of two specific project teams. In line with the distributed requirements concept, case observations were made of multiple sub-teams and stakeholder groups that were relevant for each of the projects studied. The data collection and analysis was undertaken with the intention of developing a rich understanding of how distributed requirements processes are managed on a regular basis. The case inquiries were conducted in accordance with all prevailing case study field procedures, including the development of a case study protocol prior to data collection, the use of multiple sources of evidence, and the maintenance of a chain of evidence (Yin, 2003). Specific sources of evidence included one-on-one interviews with project stakeholders, direct observation of work practices and project meetings, and review of documentary artifacts (e.g., project charters, specifications documents, interface mock-ups).

Systems Development Cases

While space does not allow for a thorough discussion of the two cases studied in this research, a brief summary is certainly warranted. The first case focused on a requirements process associated with the acquisition, customization, and implementation of an enterprise-level student information system at a mid-sized Midwestern university. Specifically, the case centers on the customization and implementation of Oracle’s PeopleSoft Student Information System (SIS) module. Several distinct roles and responsibilities were identified at the initiation of the SIS Project. An executive steering committee and executive sponsor position were established to provide oversight of the entire initiative. The executive steering committee was made up of the leading financial and administrative officers of the university as well as the lead members of the project team. Leadership of the internal project team consisted of a Project Director, three project leads (i.e., covering Functional, Technical, and Project Management domains), multiple functional leads, and a training team. The Functional Leads were responsible for coordinating the input of multiple functional subject matter experts (SMEs). Finally, a training team was tasked with the design, scheduling, and delivery of training programs to all university user groups. In addition to the internal Project Team, the university engaged the services of a consulting firm, CedarCrestone, which specializes in enterprise system implementation within the higher education marketplace. The consulting team adopted a structure directly mirroring that of the university’s Project Team. The CedarCrestone Project Manager was appointed to oversee all project management functions in conjunction with the Project Director. A CedarCrestone Account Manager worked directly with the executive steering committee and made regular recommendations to the Project Director and the CedarCrestone Project Manager.

The second environment examined was that of multi-party project aimed at establishing an information sharing platform for multiple members (i.e., agencies) of the law enforcement community within a large county on the east coast of the United States. In the Integrated Public Safety Initiative (IPSI), a regional software vendor, Blue Systems, Inc. (BSI), was selected to provide the information sharing platform that formed the core of the initiative. In this capacity, BSI professionals acted as the primary managers of the overall project effort. Specializing in public safety software, the firm had made its reputation as a developer of operations and analytical support systems for local police and fire departments. The core law enforcement agencies engaged on the initiative included the County Prosecutor’s Office, the County Sheriff’s Office, the County Corrections Department (responsible for the management of correctional facilities), and the largest municipal police department in the county. At the time of the project’s initiation, it was envisioned that all of the other municipal police departments in the county would migrate onto the platform over the subsequent three-year period.

---

1 Additional detail on both of the development environments evaluated can be made available to interested researchers or reviewers.
2 All names and acronyms associated with the project have been altered to protect the anonymity of study participants.

---

Data Analysis

Our data analysis efforts focused on a thematic analysis (Aronson, 1994; Boyatzis, 1998) of the various sources of data, including transcripts of interviews, field notes from observation sessions, and documentary artifacts from both cases. The thematic analysis was completed using Atlas.ti, a qualitative analysis application. The interview protocol served as the preliminary coding structure for the data. However, additional codes were created as specific themes or recurring issues began to surface in the coding process. The code structure was iteratively revised until the researchers determined that all relevant themes or issues were reflected (Eisenhardt, 1989). Several of the data sources were coded repeatedly as the final coding structure emerged. The aim of this analysis was to identify distinct patterns in requirements processes as well to observe emerging themes and issues in the conduct of the projects analyzed. Building upon the analysis of the two cases, we have developed an initial theoretical framework of distributed requirements processes in contemporary design environments. Next, we discuss the theoretical model that we have developed.

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

Design Requirements

The first element of the theoretical model 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 the 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.
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 observed, 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. In the two cases that we analyzed, examples of task focus included division of labor into such areas as project management, IT architecture, module development, and training.

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. (2009), including the interdependent complexity 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 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. Distribution based on domains of use was clearly observed in both of the cases studied in the development of the present theoretical framework. In the university case, examples of the domains of use included admissions, registration, and faculty domains, as well as those associated with specific schools within the university. In the law enforcement case, domains of use reflected the distinct activities of the separate law enforcement agencies (i.e., prosecutor’s office, sheriff’s office, corrections department, and municipal police). Each of these domains had distinct requirements and dedicated project team personnel were responsible for the articulation and management of requirements from each domain.

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. Examples from our cases analyses included personnel dedicated to management of requirements associated with the university’s legacy student information system and the municipal police department’s computer-aided dispatch and records management system (CAD/RMS).

Constraint sources

Finally, a significant number of constraint 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. Concrete examples of constraint sources in the cases studied included university guidelines for the protection of student records and regulatory demands for the timely prosecution on criminal suspects (e.g., prosecution of suspects must be initiated within 72 hours of an arrest).
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., 2009). 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 elements embody a range of requirements that must be aligned for the specific demands of a focal development effort. Key sources of requirements from existing platforms in the cases analyzed included the Oracle PeopleSoft platform and the BSI information sharing platform.

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. Examples of systems models employed in the two cases included use case models, entity-relationship (E-R) diagrams, and business process models.

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). Key examples from our case analyses included whiteboarded mock-ups and interface prototypes. 3

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. Obvious examples of descriptive text include the language used in the specifications documents of the two cases analyzed.

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

---

3 Interestingly, the project teams in both cases used the screen-sampling software Snagit in their interface prototyping efforts.
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., 2009). 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). The university project provided some excellent examples of temporal distribution of requirements processes, including the intensive use of external consultants (i.e., CedarCrestone) 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. 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

---

4 This example also illustrates the dynamic interplay of generative and absorptive distribution processes.
from one of the cases studied 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 distribution processes, the iterative movement between social and structural representational states is achieved. We noted above that the temporal distribution of requirements efforts is driven by the interplay of the social and structural forms of distribution. This temporal distribution is accomplished through the adoption of generative and absorptive processes. Importantly, in the day-to-day practice of requirements engineering, the movement between generative and absorptive processes is very dynamic. Accordingly, it can be difficult to discern what the primary character of a given process is. In our analysis, we argue that these are distinct facets because of a clear focus on generation or absorption of explicit requirements in a variety of the activities observed.

CONCLUSION

In this study, 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? As systems development efforts become increasingly distributed across geographic, organizational, and temporal boundaries, we anticipate that these and other questions will motivate significant research in the requirements domain.

REFERENCES


