

DESIGN KNOWLEDGE AS A LEARNING RESOURCE

Steven R. Haynes

College of Information Sciences & Technology

Penn State University

shaynes@ist.psu.edu

Abstract

Complex information systems are increasingly interwoven into the social fabric within which people learn. The process of information systems design, also inherently social, involves people from different disciplines and with different core concerns interacting to decide how purpose, structure, behavior, use, and constraints will together determine how an information technology will be created to support work (or recreation) in some domain. This paper articulates a perspective of information systems design, both the process and its artifacts, as underutilized resources for active learning. The framework draws on theories of social-constructivist learning, knowledge inscription in information systems design, and reflective practice to develop a research agenda in information systems design knowledge management and design knowledge in use. The paper positions the systems development process as an important target for knowledge management and organizational learning research, articulates a theoretical framework, examines the framework in the light of one case study, and sets a research agenda for investigating relevant issues.

Introduction

One of the goals of an evolved science is to act as a resource for learning. Physical, natural, and social scientists carry out research not only to provide knowledge for other researchers to progress their fields, but also for the general public to help them better understand and positively influence the world they live in. Much of this work involves investigating very complex phenomena, developing causal and other explanatory theories, and communicating the results of these efforts in a form usable by educated lay people and students. This broad dissemination of usable and useful knowledge is among the key missions of the scientific enterprise.

I argue here that as information systems (IS) become increasingly interwoven into the fabric of daily life, developing a program of science that acknowledges the importance of IS knowledge transparency is becoming a professional imperative. Making the mysteries of complex, software-intensive systems more comprehensible presents significant challenges, especially for a relatively new field of inquiry. The science of information systems design, development, and use is made up of an enormous range of interacting socio-technical factors. The causal force of most of these factors is either unknown or described so far not by laws, but by statistical-relevance relations that really only hint at what any one factor might mean in any given situation. To gain credit and credibility as a discipline requires finding ways to express the knowledge that we do possess so that it can be communicated and used productively by society.

This paper focuses on one small aspect of this objective: the users of complex information systems and the domain knowledge inscribed in the design of the systems they use. As society's aggregate knowledge is increasingly packaged into 'black boxes' of software-intensive information systems, it become less accessible to those without specialist skills. Even

for those with these skills, the design deliberations that lead from a problem statement to a working system are often lost as software engineering abstractions are created from the inherent disorder of a complex domain (MacLean, Young et al. 1996). These lost deliberations constitute a missed opportunity for learning as they describe how a design team arrived at particular generalizations and abstractions from analysis of the problem domain.

The knowledge embedded in complex system designs is of value not only as a resource for science and learning, but as an economic good in its own right. Information technologies including software systems are increasingly visible as a significant proportion of the U.S. gross domestic product (Mandel 2002). What is less visible is the value of the knowledge that is generated in the activities of design and then *inscribed* into these technologies and systems. Very little research has investigated the potential value of this inscribed knowledge, though some work suggests that there is real, significant value associated with the knowledge generated from design activities (Reich 1995). The program of research set out here is focused on promoting recognition of the value embedded in systems designs, and on investigating methods and tools for more effectively capturing, representing, communicating, and reusing this knowledge.

Knowledge management (KM) is concerned with the ability of organizations to make effective use and reuse of their intellectual assets (Alavi and Leidner 2001). This knowledge is increasingly vested in the information technologies that organizations create, acquire, and use. Design knowledge management is both more specific and more narrowly defined than general KM. Many of the challenges found to impede organizational KM efforts apply to the design context, and design as a relatively well-bounded organizational activity may provide a potentially fruitful sub-domain for developing and evaluating knowledge management tools and techniques for more general use.

The rest of this paper attempts to articulate a program of research to support development of a communicable science of information systems design. The first section briefly reviews social conceptions of learning and how learning is described as occurring in communities of interest and communities of practice inhabited by both people and artifacts. The next section explores the mechanisms through which one type of artifact, information systems, are inscribed with much of the knowledge germane to a community's purpose, structure, and behavior. Then, I describe an example project from the domain of defensive AT planning and attempt to show how this case exemplifies mechanisms and opportunities for leveraging the knowledge reified in a complex information systems design project. Before concluding I set out some of the key challenges and opportunities to design knowledge capture, representation, transfer, and use.

Social Learning

Social theories of learning describe learning as the result of an elaborate and nuanced relationship between learners and their environment, especially their social environment (Bandura 1977; Vygotsky 1978). Recent work on communities of practice and communities of interest shows how learning is often situated in "authentic" social situations characterized by interaction with the environment and with other people (Lave and Wenger 1991). Piaget's influential work on social-constructivist theories of learning put significant weight on the role of objects at-hand, and discussion with others about these objects, as devices to support learning about the world. Papert's work with children and the LOGO programming language is focused on how children develop and test theories about the world by inscribing these theories into working software programs (Papert 1993). What is less clear is whether and how this knowledge, once embedded in an 'invisible' computer program, can be made available to subsequent users of that program.

Some modern conceptions describe learning as a process involving the active participation of individuals in a *community of practice* (Lave and Wenger 1991). It is their

membership in and identification with a social network that provides the setting and structure for learning about how a community perceives the environment and the activities that are central to the community's existence. Active participation in these activities within the social system of a community of practice is the mechanism through which learners acquire understanding. An individual moves from novice to expert within a community of practice by becoming an increasingly active and fluent conversationalist in the community's areas of core interest.

At work and increasingly in recreation, information systems are among the most central artifacts helping to define a community of practice. Lave and Wenger claim that becoming a participant in a community of practice involves "engaging with the technologies of everyday practice" and that these technologies "carry a substantial proportion of that community's heritage" (Lave and Wenger 1991). They are also among the most significant of the structural constraints that determine the boundaries of a community's possible activities, the business or other rules reified in an information system are the result of a community's negotiations about how best to support their activities in a domain. Deep understanding of an information system at the center of a community is often a mark of advanced status and expertise within the community.

The idea of designers as *reflective practitioners* provides a theoretical anchor for the *value* of studying design knowledge and design knowledge interaction as the basis for a science of design focused on learning (Schön 1983; Schön 1987). This view describes design as a *conversation with materials* (Schön and Bennett 1996), where people interact with each other and with the materials of their design efforts as means to understanding a domain and for formulating interventions within it. Understanding the nature of these materials and conversations about their properties in light of problems and solutions is seen as critical to becoming an enlightened practitioner in professional fields. For many knowledge workers, information systems are the materials they use to design solutions in their field of practice. That the attributes of these materials are invisible to them creates a barrier to evolving their domain understanding and professional development.

Information System Designs as Knowledge Stores for Social Learning

Though there has been substantial research on learning *about* computers and software systems (Card, Moran et al. 1983; Norman 1988; Carroll 1990), less work has explored the design implications of supporting people as they learn from their increasingly pervasive, day-to-day use of information technology. Conceptions of information systems as *cognitive artifacts* acknowledge that devices and tools in our environment represent knowledge about how to perform tasks in their intended domain (Norman 1991). This acknowledged role of tools as cognitive prostheses is relatively static in its delineation of responsibilities however, and does not admit the possibility that people can learn and evolve their domain understanding through active engagement with the knowledge inscribed into their design (Kaptelinin 2003). Others have pointed out that there is in fact just such an interplay of people and technology in the performance of activities (Lave and Wenger 1991), but the mechanisms of how people might learn from technology interactions is generally opaque in this work.

A range of theoretical positions and empirical findings suggest that people and artifacts operate as an active network where people make use of their own knowledge, the knowledge of others, and the knowledge or *prior cognitions* embedded in the technologies they employ (Latour and Woolgar 1986; Hutchins 1995). Technologies emerge from design activities, which have been characterized as *moves* within a *design space*. These moves include forming intentions (Elster 1983; Leveson 2000); evaluating use scenarios (Carroll 2000); deliberating on the questions, solution options, and selection criteria that arise from them (MacLean and McKerlie 1995); considering design constraints (Vincenti 1990); among many others. All these moves involve the collection and skillful application of design knowledge. It is widely acknowledged,

however, that the design process is characterized by a certain entropy and that much of the knowledge that surfaces during design is lost to those who might later use it to understand built systems, adapt and reuse the problem-solving approaches inscribed in these systems, and gain an appreciation for salient aspects of the domains for which these systems are designed.

Prior cognitions are resident in human long-term memory; in the structure, behavior, and data that make up an information system; and in the specifications, source code and comments, and supporting materials such as documentation, help systems, and policies constructed to support an information system in use. These prior cognitions represent a significant knowledge base describing a system's operational domain and the domain activities the system is designed to support. Harnessing this knowledge in service of individual and organizational learning requires first of all acknowledging its inherent value, and then developing strategies, techniques, and tools for knowledge capture, representation, and access.

In this paper I focus on a particularly rich source of prior cognitions, that is, the information systems design and development process. Activities such as business analysis, requirements analysis, systems design, construction, test, implementation, and support all act to reify, albeit all too briefly, a vast body of knowledge regarding an organization's 'posture' with respect to some operating domain such as the supply chain, executive decision making, or engineering project management. However, our ability to leverage the knowledge generated from these activities is notoriously underdeveloped.

One long acknowledged challenge to comprehending software-intensive information systems is their invisibility relative to the domain activities they are designed to support (Brooks 1987). The full depth of knowledge is generally not made explicit in design inscriptions. Information systems designed to support collaborative, distributed knowledge work are, with their lack of physical structure cues to help understand their design, even harder to consider with respect to the design space from which they emerged.

Tools and technologies embed a wide range of prior cognitions about the context in which they are intended for use, about the tasks for which they are designed, and about the kinds of constraints that led to a particular realization of the idea. The cultural and temporal breadth of these prior cognitions are exemplified in Petroski's work on the origins of the dinner fork, which shows how this very simple artifact inscribes an astonishing array of historical-cultural content (Petroski 1996). Their depth is discussed for example in Vincente's accounts of how various devices engineered for aeronautics emerged from the intense efforts of a well-resourced and motivated community of engineers and scientists (Vincenti 1990).

The work of Edwin Hutchins (Hutchins 1995; Hutchins 1995) provides perhaps the most complete explication of the role of inscriptions as prior cognition. In his study of the aircraft landing task, he describes how the complex task of determining appropriate landing speeds for different aircraft configurations is pre-computed and represented in the aircraft cockpit as a book of tables. Pilots employ these tables in a simple look-up task which replaces what would otherwise be a time-consuming and cognitively demanding sequence of steps. In this case, knowledge of the aircraft structure and behavior, axioms of aeronautics, and data on the effects of environmental factors (e.g., wind) are distilled into an accessible and easy to use knowledge base.

Knowledge is inscribed into information systems in at least three different ways. First, there is the core design, the algorithms and data structures that make up the software system. Second is the parameter or standing data that dictates how the system will behave within its operational boundaries. The SAP application suite for enterprise resource planning, for example, includes thousands of configuration switches, parameter data, that interact to control exactly how the system will function and be operated. Organizations often pay substantial sums of money for the expertise needed to set these parameter data so that the system conforms to their business

model. Finally there is system content, the information that is captured, processed, and stored to fulfill the system's functional requirements.

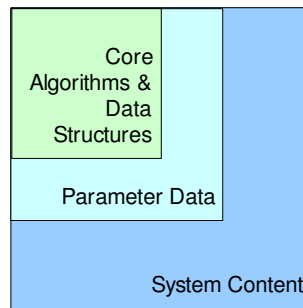


Figure 1 - Knowledge Types Incribed in Information Systems

This paper focuses on the first two of these three knowledge kinds, the structures and behaviors created during design rather than from a system in use, and, importantly, the reasons or rationale for how and why a given system possesses the structure and behaviors that it does.

An Example from Anti-Terrorism Planning Systems

The *Rampart* anti-terrorism planning system is the product of a three-year collaboration between the United States Marine Corps and a research team from academia. Rampart is a decision model realized in a cognitive support system designed to be used by AT officers, facilities planners, public works officers, and emergency response personnel. The model and system provide support for asset prioritization, calculation of AT mitigation project utilities, and resource allocation. The system also acts as a repository and resource for organizational learning in the AT domain.

Rampart is essentially a decision support system. Users of the system work through a number of planning tasks to build a model describing an infrastructure to protect. They identify prioritization criteria, weight these criteria, identify the different infrastructure assets to be protected, weight these against prioritization criteria, and then design mitigation projects that might be applied to identified assets, for example, stand-off barriers (the mitigation project) placed around a tank farm (the asset) at an oil refinery (the infrastructure). The goal of the system is to identify the most effective combination of mitigation projects given a constraint, usually money, but potentially people, equipment, or some other scarce resource.

Rampart emerged from extensive fieldwork with the Marines Corps; one strength of the system is that its design is grounded in results derived from observing and interacting with AT planners and other Marine Corps personnel concerned with how to allocate necessarily limited resources to those AT projects and activities with the most benefit. In the course of the three-year study the project team carried out requirements elicitation, prototype design reviews, focus groups, and structured walkthroughs with over 200 people from the Marines as well as the U.S. Army Corps of Engineers, the U.S. Navy Facilities Command, representatives from various agencies within federal and state government, and civilian emergency planning personnel including government contractors.

In the course of these research and design activities we learned a great deal about the defensive AT planning domain including the operation and protection of military installations, the nature and use of AT technologies and other mitigation techniques, terrorist event consequence planning and management, as well as about the organizations and people engaged in the task of AT planning and how these activities with a relatively new high profile were being incorporated into day-to-day operations. As academics, we also carried out research in fields that we believed

would help inform our efforts including normative and naturalistic decision making, human-computer interaction with decision systems, software engineering and systems development, and related works from our own and other fields on AT planning and emergency management.

Much of this knowledge was subsequently inscribed into the design of the cognitive support system we constructed. We continued to refine the design in response to feedback we received from prospective users who participated in one-on-one guided walkthroughs. The version of the system at projects end was, based on results of final walkthroughs and design reviews, a reasonably complete and high quality prototype system representative of the ‘stance’ of the Marine Corps towards the defensive AT planning domain.

In these later interactions with prospective users and other project stakeholders we were continually challenged to provide the reasons for why the system addressed specific domain activities in a particular way. How and why, for example, had we selected a particular approach to considering people versus mission priority when prioritizing different installation facilities. Senior military commanders wanted to understand the thought process, the knowledge, that went into making a decision on how to handle sensitive problems such as these. Some examples of issues where such questions arose about designs features in the Rampart system, along with the knowledge underlying how they were realized appear in Table 1 below.

Table 1 - Design Features and Underlying Knowledge in Rampart

Design Feature	Underlying Knowledge
Asset prioritization	Ranking algorithms - AHP, SMART and their trade-offs Normative decision making Cognitive psychology Naturalistic decision making Military infrastructure Mission-essential assets
Mitigation project utility	Anti-terrorism mitigation effectiveness Mitigation costs Social valuation of human life
Anti-Terrorism resource allocation	Linear programming algorithms Decision sensitivity analysis

Though the system design includes a training module and complete help system, and studying the use of these was an important part of our research project, the content used to populate these tools does not include the rationale for every decision made as part of the design process including how we identified candidate solutions, selected from between them, and the criteria we used in these decisions. As time goes by much of what we learned and inscribed into the design of this system will be lost, both to us and to the users of the system who are already distanced from it by the organizational boundaries between us. This is a familiar problem in

systems development, in requirements tracing, in system documentation and training, and in design rationale. A learning resource from an important and timely domain is being lost to those who sponsored us to help with what for them is a day-to-day concern.

Challenges and Opportunities

Many challenges to leveraging design knowledge as a resource for a learning, and for an evolved science of design, relate to design knowledge *capture* (including representation and transfer) and design knowledge *in use*. In the two sections that follow I elaborate on these challenges and describe how their consideration leads to some fundamental questions that must be addressed to support employing design knowledge as a resource for learning.

Design Knowledge Capture

Much of the knowledge surfaced in the software-intensive systems design process, for example, in design meetings, in conversation and e-mail, and by designers and developers working independently, is not captured for use and later reuse. Information captured about the decisions made in systems design typically consists of only the course-of-action decided, not the rich knowledge content emerging from the individual and group deliberations that led to the decision. This body of information has been described as a *design space* (MacLean, Young et al. 1996) and the tools and techniques developed to help capture, represent, and use this information are part of an approach to design known as *design rationale* (Moran and Carroll 1996). A design space represents an organization's stance or posture towards some problem domain. As a design team 'moves' through a design space they surface a wealth of knowledge to resolve emerging design questions. Capturing the full breadth and depth of a design space and effectively using and reusing this knowledge promises a number of benefits, but realizing these benefits poses significant social, scientific, and engineering challenges.

A central problem with design space knowledge capture is preserving the fluidity of the design process while at the same time capturing design-relevant deliberations wherever or whenever they occur. Addressing issues of design knowledge capture requires development of usable, pervasive capture strategies and tools. Design knowledge capture should be non-intrusive and preserve individual privacy and at the same time be capable of capturing all the valuable knowledge that is reified, however briefly, in design deliberations. Design capture technologies should be easy to use with minimal training, they should be mobile, relatively inexpensive, and integrated with existing tools and representations. Crucially, they should provide user services that justify and provide incentives for use.

Schön's conception of the reflective practitioner, described earlier, is one of the foundations of the design rationale approach to capturing, representing, and reasoning about design knowledge. Moran and Carroll distinguish between the process and product of a given design effort (Moran and Carroll 1996). Typically, when a design effort has been completed, organizations are left with the products of design including the designed artifact and a set of documentation that describes the artifact in highly variable levels of detail and quality. They argue that this documentation rarely includes a complete record of the initial motivation for the project including project team and external stakeholder deliberations and negotiations, and the reasons that were finally applied to justify the inclusion and form of particular system features. It has been argued that to achieve understanding of a system means making transparent the concepts that underlie its architecture (Wenger 1987). In design rationale, these concepts are captured and expressed as the "argument behind the artifact" (MacLean, Young et al. 1989).

Cross claims that the design movement has progressed through four distinct phases that he identifies as prescription, description, observation, and reflection (Cross 1984). Design

rationale may be characterized as an outcome of this fourth phase. Design rationale is concerned with the entire design space of a project rather than exclusively with the artifacts that emerge from separate design activities (MacLean and McKerlie 1995). The arguments and deliberations that occur within this design space and the justifications that underlie the form and function of the artifacts that emerge from it are of central importance. Use of design rationale is typically combined with other project artifacts, methodologies, tools, and development philosophies to provide a more holistic representation of the design knowledge ‘ecology’ for a given project (Potts and Bruns 1988).

A range of representation strategies and associated tools has been developed to support capture and reasoning with design rationale. Though they all follow similar node-typed link structures (Shipman and McCall 1996), these formalisms vary in terms of how structured they are with the acknowledged trade-off being increased tractability by users versus increased tractability by machines (Lee and Lai 1996). The Issue Based Information System (IBIS) (Rittel 1984) is perhaps the earliest design rationale representation. Elements of a problem domain are represented by three node types: *issues* are identified for which stakeholders take *positions*; these positions are backed by *arguments*. The Procedural Hierarchy of Issues (PHI) another IBIS-based design rationale formalism (McCall 1991) attempts to solve a number of key deficiencies in the original IBIS framework. In the Questions-Options-Criteria (QOC) notation (MacLean, Young et al. 1996), questions highlight issues that have been identified as relevant to the design, options are the potential solution approaches that have been identified to address a given question, and criteria are the reasons that are considered for or against each of the identified options.

Despite the level of research activity directed at problems associated with capturing and representing design knowledge in the form of rationales, relatively little of this work involved empirical study and those that have been carried out point to significant issues with prescriptions for design capture. A series of lab studies of design rationale usability (Buckingham Shum and Hammond 1994; Buckingham Shum 1996; Buckingham Shum, MacLean et al. 1997), for example, identified significant issues with the usability of the QOC notation. Their analysis identifies some of the points at which breakdowns occur in the design rationale capture process, but no work has yet attempted to identify notational, representational, and tool affordances that might mediate some of these usability issues.

As a complex and necessarily social activity, issues with supporting design knowledge are conjectured to reflect many of the findings from research in computer-supported cooperative work. Ackerman (Ackerman 2000) characterizes many of these issues as arising from the essential gap between the complex subtleties of human social interaction and the capabilities of existing technologies to support them. Among the many challenges that emerge from this social-technical gap is the disjunction between the need for almost constant, rigid categorization of design information, and the human ability to treat categories and concepts as fuzzy, adaptable, and evolving. This lack of flexibility when dealing with often ambiguous information is especially salient to design knowledge management and especially design knowledge capture, where designers are asked not only to apply their knowledge effectively to design tasks but also to reify and taxonomize this knowledge as they work through difficult design problems.

Grudin’s studies and analyses of groupware implementation and use are among the most influential guides to successful, and failed, collaborative system implementations, such as those inevitably required to support design knowledge capture (Grudin 1988; Grudin 1994). He identified eight key challenges including: disparity between users, achieving critical mass, disruption of social processes, requirements for exception handling, ease of accessibility, problems with evaluation and redesign, management “failure of intuition”, and challenges to the system adoption process. The first four of these factors especially highlight the kinds of factors

developers of design capture systems might face when attempting to implement in organizational contexts.

There have however been some positive results from studying design rationale in the field. In applying IBIS-based design rationale over an extended period on a design project, Conklin & Burgess-Yakemovic (Conklin and Burgess-Yakemovic 1996) found that design rationale both improved design meetings by providing an agenda and capturing results of previous meetings and assisted in the process of acclimating new team members to a project. This is perhaps the only longitudinal field study of design rationale effectiveness.

Information systems have been described as representing “frozen organizational discourse” (Bowker and Leigh-Star 1994), conveying a stance toward activity in some domain that embodies current thinking about the best way to work in the domain. As described above, not all of this knowledge is explicit in these inscriptions, and software-based information systems—with their lack of physical structure and other cues to help understand their design—are even more difficult to comprehend *prima facie* with respect to the design goals from which they emerged. This embedded knowledge receives little attention from its owners. Part of this is that the embeddedness and invisibility make the knowledge difficult to probe. In the next section I describe a research program for design knowledge interaction that may help to address this and some related issues.

Interacting with Design Knowledge

As a field of study, interactions with design knowledge integrates prior work from reference disciplines such as software engineering, design rationale, human-computer interaction, computer-supported cooperative work, and knowledge management. Integrated perspectives such as this are increasingly recognized as necessary to the continued development and success of the computer and information sciences as they provide more holistic and ecologically valid bases for inquiry and understanding (Carroll and Rosson 2003).

Among the prerequisites to supporting making use of design is obtaining an analytic understanding of the essential nature of design knowledge interaction. This may involve, for example, developing cognitive task analyses (Annett and Stanton 2000) of the activities associated with designing complex, software-intensive systems and integrating these with results of prior research on design knowledge interaction (Jeffries, Turner et al. 1981; Herbsleb and Kuwana 1993; Buckingham Shum and Hammond 1994). The objective of these research activities is to develop a single representation to serve as a baseline model for more empirical studies.

Also required is infrastructure to serve as a research apparatus for empirical studies of design interactions. Especially important is creation of a design knowledge case base capable of serving as both a content repository and research test-bed for empirical studies of design knowledge interaction. Existence of this infrastructure will enable semi-structured, exploratory studies such as cognitive walkthroughs with interaction experts (Polson, Lewis et al. 1992) and guided walkthroughs (Haynes, Puro et al. 2004) with prospective users of design repositories, for example, system architects, developers, users, and project stakeholders. The purpose of such exploratory studies is to help discover early usability issues with design knowledge interactions, and to suggest innovative approaches to support interacting with design.

Exploratory studies such as those described above require more extended, ecologically valid research to help tease out design interaction issues that arise in organizational contexts. Very little research has been performed to study complex systems design knowledge interaction, and the technology affordances that might best support these interactions. Field studies serve as test cases for the theoretical propositions emerging from earlier activities and the design case base

developed to support interaction research. In this phase of the research, particular attention is paid to identifying specific hypotheses that can be tested in more controlled, experimental studies as described below.

Provided with tested research tools and results from exploratory and field studies, researchers may then begin more controlled, laboratory study to investigate specific constructs and hypotheses. This program of iterative design research is based on a commitment to using ecologically valid, but necessarily less rigorous, field studies as the basis for identifying independent variables, dependent variables, and threats to validity that are used in the design of experiments. Especially in fields such as design knowledge interaction, where very little research has been done and very little is currently known, combining qualitative and quantitative approaches is acknowledged as an effective means to progress toward understanding domain phenomena.

Remenyi and Williams (Remenyi and Williams 1996) describe how programs of research such as that described so far may lead to development of scientific paradigms. Qualitative exploratory and field research can lead to development of specific hypotheses to be tested in controlled settings, ultimately leading to development paradigmatic thinking within a research community. Their continuum is shown in the figure below.

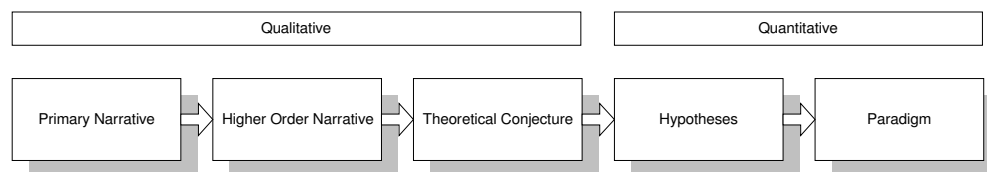


Figure 2 - Qualitative to Quantitative Research Continuum

Though paradigmatic thinking is sometimes considered a *barrier* to progressive science, especially as these paradigms grow stale and begin to actually inhibit innovative research (Kuhn 1996), for fledgling science they serve to provide a reference model through which the results from otherwise disconnected research programs can be integrated for more holistic understanding of a problem domain.

Conclusion

A major barrier to developing a scientific understanding of software-intensive systems, and to supporting effective use, is the invisibility of the underlying structure of these systems relative to the functions that they perform. Users and other stakeholders of a system do not necessarily understand the process by which these abstract systems are derived from the concrete use contexts they do understand and that make up the domain in which the system is intended for use. Scenarios of use act as a starting point for system design, but as analysis, design, and construction progresses the design that evolves to realize these requirements is molded and distorted by a diverse array of factors including the material constraints of hardware and software performance, the psychological constraints of cognition, organizational constraints of time and other resources, and social constraints imposed by standards bodies and accepted professional best practice.

The psychological and social processes that most directly impact the task of capturing and using the intellectual products of design are poorly understood, as are the technology affordances that are required to support them. The program of research proposed here focuses on a particular aspect of design knowledge management: those phenomena that emerge at the

interface where people use computers and other technologies to capture, manipulate, and reuse the products of design.

Approaching design knowledge management requires that we understand a network of factors within which people, technologies, and other artifacts interact to understand, structure, and design solutions to important problems in complex domains. A single study finding or single technological development is unlikely to resolve the critical issues attending design knowledge capture and use. A science of design centered on design knowledge management might take as its core a number of essential topics including among others:

1. How can we most effectively study design and designers?
2. How can design knowledge management be integrated into the education and training of future software system design professionals?
3. What representations can contribute to resolving the tension between ease of use and computer tractability?
4. What computer and audio-visual technologies might afford effective design capture?
5. What breakdowns occur in design knowledge capture and how can these be mitigated with process, notation, and technology support?
6. How can a mass of design information be transformed into useful, usable, and reusable domain knowledge?
7. How can design knowledge be most effectively leveraged to improve the cost-benefit of capture through new and innovative applications — what is the up side of design capture?

This paper has discussed a number of interventions to help capitalize on the scientific possibilities inherent in complex information systems design. These include finding new processes, representations, and tools to support design knowledge capture, and finding the same to support more effective interactions with the knowledge products of design. I have argued that design knowledge management is central to the development of a science of complex information systems design, and that making design knowledge more transparent both to system users and society at-large is a professional imperative. Significant challenges stand in the way of achieving success in the research program, not least of which is a certain inertia created as a result of prior research on information system development and use, in design knowledge capture, and in the difficult problems associated with fostering active, reflective learning from the products of design.

Acknowledgements

This research was supported by a contract with the United States Marine Corps through the Marine Corps Research University and by the Office of Naval Research under grant number N00014-06-1-0164.

References

- Ackerman, M. (2000). "The Intellectual Challenge of CSCW: The Gap between social requirements and technical flexibility." *Human-Computer Interaction* **15**: 179-203.
- Alavi, M. and D. E. Leidner (2001). "Knowledge management and knowledge management systems: Conceptual foundations and research issues." *MIS Quarterly* **25**(1): 107-136.

- Annett, J. and N. Stanton (2000). Task analysis. London; New York, Taylor & Francis.
- Bandura, A. (1977). Social learning theory. Englewood Cliffs, NJ, Prentice Hall.
- Bowker, G. and S. Leigh-Star (1994). Knowledge and infrastructure in international information management: Problems of classification and coding. Information Acumen: The understanding and use of knowledge in modern business. L. Bud-Frierman. London, Routledge: 187-216.
- Brooks, F. P. (1987). "No silver bullet: Essence and accidents of software engineering." IEEE Computer **20**(4): 10-19.
- Buckingham Shum, S. (1996). Analyzing the Usability of a Design Rationale Notation. Design Rationale: Concepts, Techniques, and Use. T. P. Moran and J. M. Carroll. Hillsdale, NJ, Lawrence Erlbaum Associates: 185-215.
- Buckingham Shum, S. and N. Hammond (1994). "Argumentation-Based Design Rationale: What Use at What Cost?" International Journal of Human-Computer Studies **40**(4): 603-652.
- Buckingham Shum, S., A. MacLean, et al. (1997). "Graphical Argumentation and Design Cognition." Human-Computer Interaction **12**(3): 267-300.
- Card, S. K., T. P. Moran, et al. (1983). The psychology of human-computer interaction. Hillsdale, N.J., L. Erlbaum Associates.
- Carroll, J. M. (1990). The Nurnberg Funnel. Cambridge, MA, MIT Press.
- Carroll, J. M. (2000). Making use: scenario-based design of human-computer interactions. Cambridge, MA, MIT Press.
- Carroll, J. M. and M. B. Rosson (2003). Design Rationale as Theory. HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science. J. M. Carroll. New York, Morgan Kaufmann: 431-460.
- Conklin, E. J. and K. C. Burgess-Yakemovic (1996). A Process-Oriented Approach to Design Rationale. Design Rationale: Concepts, Techniques and Use. T. Moran and J. M. Carroll. Mahwah, NJ, Lawrence Erlbaum: 393-427.
- Cross, N. (1984). Developments in design methodology. New York, Wiley.
- Elster, J. (1983). Explaining Technical Change: A Case Study in the Philosophy of Science. Cambridge, UK, Cambridge University Press.
- Grudin, J. (1988). Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. Proceedings of CSCW '88, ACM Press.
- Grudin, J. (1994). "Groupware and Social Dynamics: Eight Challenges for Developers." Communications of the ACM **37**(1): 92-105.
- Haynes, S. R., S. Purao, et al. (2004). Situating evaluation in scenarios of use. Proceedings of the 2004 ACM conference on Computer supported cooperative work, Chicago, Illinois, ACM Press.
- Herbsleb, J. D. and E. Kuwana (1993). Preserving Knowledge in Design Projects: What Designers Need to Know. Proceedings of INTERCHI'93, ACM Press.
- Hutchins, E. (1995). Cognition in the wild. Cambridge, Mass., MIT Press.
- Hutchins, E. (1995). "How a Cockpit remembers Its Speeds." Cognitive Science **19**: 265-288.

- Jeffries, R., A. T. Turner, et al. (1981). The Processes Involved in Designing Software. Cognitive Skills and Their Acquisition. J. R. Anderson. Hillsdale, NJ, Lawrence Erlbaum: 255-283.
- Kaptelinin, V. (2003). "Learning with artefacts: integrating technologies into activities." Interacting with Computers **15**: 831-836.
- Kuhn, T. S. (1996). The structure of scientific revolutions. Chicago, IL, University of Chicago Press.
- Latour, B. and S. Woolgar (1986). Laboratory life: The construction of scientific facts (reprint edition). Princeton, NJ, Princeton University Press.
- Lave, J. and E. Wenger (1991). Situated learning: legitimate peripheral participation. Cambridge [England]; New York, Cambridge University Press.
- Lee, J. and K.-Y. Lai (1996). What's in Design Rationale? Design Rationale: Concepts, Techniques and Use. T. P. Moran and J. M. Carroll. Mahwah, NJ, Lawrence Erlbaum: 21-51.
- Leveson, N., G. (2000). "Intent Specifications: An Approach to Building Human-Centered Specifications." IEEE Trans. Software Eng. **26**(1): 15-35.
- MacLean, A. and D. McKerlie (1995). Design Space Analysis and Use-Representations. Scenario-Based Design: Envisioning Work and Technology in System Development. J. M. Carroll. New York, John Wiley.
- MacLean, A., R. Young, et al. (1989). Design Rationale: The Argument Behind the Artifact? Proceedings of the ACM Conference on Human Factors in Computing Systems, New York: ACM.
- MacLean, A., R. M. Young, et al. (1996). Questions, Options, and Criteria: Elements of Design Space Analysis. Design Rationale: Concepts, Techniques and Use. T. P. Moran and J. M. Carroll. Mahwah, NJ, Lawrence Erlbaum: 21-51.
- Mandel, M. J. (2002). "Tech's Weakness Is Only Relative: Its share was much lower in '91." Business Week: 30.
- McCall, R. (1991). "PHI: A Conceptual Foundation for Design Hypermedia." Design Studies **12**(1): 30-41.
- Moran, T. P. and J. M. Carroll (1996). Design rationale: concepts, techniques, and use. Mahwah, NJ, L. Erlbaum Associates.
- Norman, D. (1991). Cognitive artifacts. Designing Interaction: Psychology at the Human-Computer Interface. J. M. Carroll. Cambridge, MA, Cambridge University Press.
- Norman, D. A. (1988). The psychology of everyday things. New York, Basic Books.
- Papert, S. (1993). Mindstorms: children, computers, and powerful ideas. New York, NY, BasicBooks.
- Petroski, H. (1996). Invention by design: how engineers get from thought to thing. Cambridge, MA, Harvard University Press.
- Polson, P. G., C. Lewis, et al. (1992). "Cognitive walkthroughs: A method for theory-based evaluation of user interfaces." International Journal of Man-Machine Studies **36**: 741-773.

- Potts, C. and G. Bruns (1988). Recording the Reasons for Design Decisions. Proceedings of the Seventh International Workshop on Software Specification and Design. Singapore, IEEE: pp. 418-427.
- Reich, Y. (1995). "Measuring the value of knowledge." International Journal of Human-Computer Studies **42**(1): 3-30.
- Remenyi, D. and B. Williams (1996). "The nature of research: qualitative or quantitative, narrative or paradigmatic?" Information Systems Journal **6**: 131-146.
- Rittel, H. (1984). Second Generation Design Methods. Developments in Design Methodology. N. Cross. New York, John Wiley.
- Schön, D. and J. Bennett (1996). Reflective Conversation with Materials. Bringing Design to Software. T. Winograd, Addison-Wesley: 171 - 189.
- Schön, D. A. (1983). The reflective practitioner: how professionals think in action. New York, Basic Books.
- Schön, D. A. (1987). Educating the reflective practitioner: toward a new design for teaching and learning in the professions. San Francisco, Jossey-Bass.
- Shipman, F. M. I. and R. J. McCall (1996). Integrating Different Perspectives on Design Rationale: Supporting the Emergence of Design Rationale from Design Communication. College Station, TX, Center for the Study of Digital Libraries, Texas A&M University.
- Vincenti, W. G. (1990). What Engineers Know and How They Know It: Analytical Studies from Aeronautical History. Baltimore, The Johns Hopkins University Press.
- Vygotsky, L. S. (1978). Mind in Society. Cambridge, MA, Harvard University Press.
- Wenger, E. (1987). Artificial Intelligence and Tutoring Systems. Los Altos, CA, Morgan Kaufmann.