Journal of the Association for Information Systems

JAIS

Special Issue

Cognition Matters: Enduring Questions in Cognitive IS Research

Michael Davern
The University of Melbourne
mjdavern@unimelb.edu.au

Teresa Shaft
The University of Oklahoma
tshaft@ou.edu

Dov Te'eni
Tel Aviv University
teeni@post.tau.ac.il

Abstract

We explore the history of cognitive research in information systems (IS) across three major research streams in which cognitive processes are of paramount importance: developing software, decision support, and human-computer interaction. Through our historical analysis, we identify "enduring questions" in each area. The enduring questions motivated long-standing areas of inquiry within a particular research stream. These questions, while perhaps unapparent to the authors cited, become evident when one adopts an historical perspective. While research in all three areas was influenced by changes in technologies, research techniques, and the contexts of use, these enduring questions remain fundamental to our understanding of how to develop, reason with, and interact with IS. In synthesizing common themes across the three streams, we draw out four cognitive qualities of information technology: interactivity, fit, cooperativity, and affordances. Together these cognitive qualities reflect IT's ability to influence cognitive processes and ultimately task performance. Extrapolating from our historical analysis and looking at the operation of these cognitive qualities in concert, we envisage a bright future for cognitive research in IS: a future in which the study of cognition in IS extends beyond the individual to consider cognition distributed across teams, communities and systems, and a future involving the study of rich and dynamic social and organizational contexts in which the interplay between cognition, emotion, and attitudes provides a deeper explanation of behavior with IS.

***Keywords**: Cognition, Interactivity, Fit, Affordances, Cooperativity, Systems Development, Decision Support Systems, Human-Computer Interaction*

* Carol Saunders was the accepting senior editor. This article was submitted on 2nd April 2011 and went through two revisions.

Volume 13, Special Issue, pp. 273-314, April 2012

Cognition Matters: Enduring Questions in Cognitive IS Research

1. Introduction

In 1977, several landmark events occurred in the information systems (IS) discipline. *MIS Quarterly* was established as one of the first journals dedicated to the new field. *Management Science* published the "Minnesota experiments" paper (Dickson, Senn, & Chervany, 1977), since cited more than 250 times (Harzing, 2011). While *MIS Quarterly* is hallowed as a premier journal in the field, research from a cognitive perspective, such as the Minnesota experiments, has not received such broad recognition. Yet, nearly a quarter of all recipients of AIS Fellow awards[1] pursued a cognitive-related topic in their PhD program. Our purpose here is to explore the evolution of cognitive research in IS, and in so doing, identify its future directions and potential for contribution. Consistent with calls to focus on the IT artifact in IS research (Benbasat & Zmud, 2003; Orlikowski & Iacono, 2001), much cognitive research in IS has centered on the IT artifact. With a focus on the IT artifact, and history to the early days of the field, an historical analysis of cognitive research in IS provides a unique opportunity to examine the evolution of IS research with an eye to the future.

The term "cognitive psychology" was coined by Neisser (1967), who describes cognition as "the activity of knowing: the acquisition, organization and use of knowledge" (Neisser, 1976, p. 1). Cognition entails both knowledge structures (organization) and processes (acquisition and use) that occur within a given of (human) cognitive architecture (e.g., short-term vs. long-term memory, and so forth). In parallel, an IS can be viewed as an IT artifact for the "acquisition, organization and use of knowledge". Cognitive psychology is, thus, of fundamental relevance to IS, indeed it is sometimes referred to as information processing psychology.

In our exploration of past cognitive research within IS and the future possibilities it offers, we are cognizant of what Weber (2003) calls the "error of inclusion" – researching questions that are part of the reference discipline rather than IS. Our interest is the advancement of IS as a discipline, not of cognitive psychology. Therefore, while we may borrow insight, theory, and methods from a reference discipline, we must address phenomena "that are not the focus of other disciplines" or "applications of theories from other disciplines or straightforward extensions of these theories" (Weber, 2003, p. vi). Our interest is to understand IS phenomena where it overlaps with cognition. However, one could argue that nearly any IS study investigates cognition, given that information and using it seem to imply cognitive activities of some nature. Hence, we focus on studies that employ cognitive theories and explanations for IS phenomena.

What then is the scope of cognitive research in IS, and our history in particular? In defining the scope of psychology, James (1950, p.3), notes about the mind: "the faculty does not exist absolutely, but works under conditions and the quest of the conditions becomes the psychologist's most interesting task". In a similar vein, cognitive research in IS explores the interactions between cognition and context that influence behaviors and outcomes in the development and use of IS. We discuss the issue of scope further in the next section when we develop our framework for understanding cognition and information systems.

Our historical analysis is not the same as a comprehensive review (the span of cognitive research in IS would easily fill a book). Rather, we seek to identify, through retrospective analysis, enduring IS questions in which cognition plays a central role. By enduring IS questions, we mean fundamental issues that have motivated various studies regarding cognition and IS. We identify these enduring questions through our retrospective analysis and find that they provide a useful means to organize cognitive research in IS. Further, given their historical significance and long-lasting nature, they allow us to provide advice about future cognitive research in IS. As a matter of scope we concern ourselves principally with questions that have implications for the design and use of IS because these are lines of inquiry where cognition has been particularly relevant.

In the remainder of this paper we first present our organizing frameworks for exploring cognition in IS.

[1] For example: Benbasat, Ein-dor, El-Sawy, Galletta, Ives, Liang, Munro, Te'eni, Vessey, and Weber

Second, we analyze the history of cognitive research in IS for each of the three streams derived from the structure of our organizing frameworks: development, decision support, and human-computer interaction. We then synthesize and reflect on the historical contributions to IS cognitive research. We conclude with a speculative exploration of the future of cognitive research in IS by extrapolating from our historical observations.

2. A Framework for Exploring Cognitive Research in IS

2.1. Information Systems as Representations

We develop our historical analysis of cognitive research in IS with an organizing framework that draws on an established conceptualization of IS (i.e., Wand & Weber, 1990) and a widely-cited model of human information processing (i.e., Card, Moran, & Newell, 1983). Because we are interested in advancing IS research, we begin with defining our perspective on IS. We consider the IS artifact as a representation of some underlying reality (Wand & Weber, 1990; Weber, 2003). "Representation" is a critical aspect of the well-known theory of cognitive fit (Vessey & Galletta, 1991). Mental representations also play a central role in the model of human information processing that we adopt from cognitive psychology. The description of problem solving provided by the "paradigm of cognitive fit" (Vessey & Galletta, 1991, p. 66) is helpful to understanding our framework:

> *The basic model views problem solving as the outcome of the relationship between the problem or external representation and the problem solving task, which are characterized for the purposes of this analysis by the type of information emphasized … processes act on information in (1) the problem representation and (2) the problem solving task, to produce the mental representation; and (3) the mental representation to produce the problem solution. The mental or (internal representation) is the way the problem solver represents the problem in human-working memory.*

The term representation can, thus, refer to an internal (mental) representation or an external representation. The "external representation" in this context is the manner in which information is presented to a user of an information system. Figure 1 depicts our organizing framework that expands on the notion of information systems as representations in a manner consistent with the "human information processor" model (Card et al., 1983) of cognition (as discussed below).

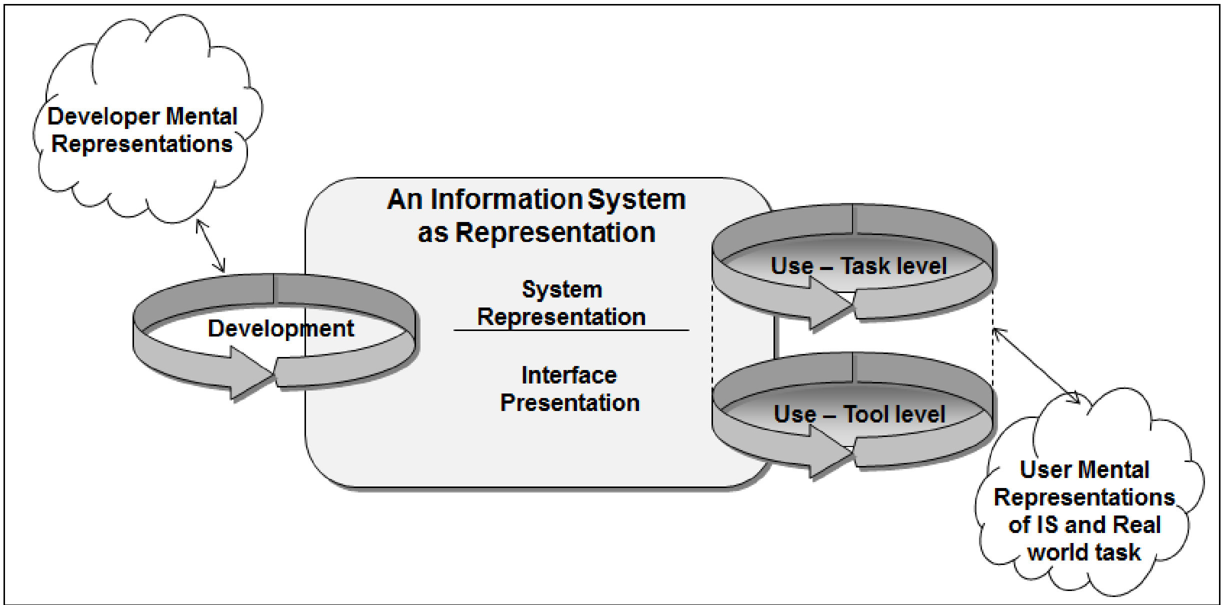

Figure 1. An Organizing Framework for Exploring Cognition with Information Systems

As shown in Figure 1, we distinguish developing an IS representation from use of a representation, although in many contexts the stages are intertwined. We further distinguish between two levels of use: the task level and the tool level (following Davern, 1996, 2007; Moran, 1981; Te'eni, Carey, & Zhang, 2007). At the task level, the System Representation (i.e., the information system as representation in Weber's sense of the term) is comprehended, manipulated, and tested to reason, solve problems, and act. For example, a task level activity could be consumer decision making, an activity that has proliferated in studies of online shopping. At the tool level, the (computerized) tool is used to view and manipulate the "presentation" of the underlying system representation to support the task level: for example, online graphics or color (tool level) to support decision making (task level). In addition, there are the corresponding internal mental representations [2] of the users and the developers (shown in the clouds in Figure 1). From the perspective of distributed cognition, there are also representations in the minds of others in the social context (Hutchins, 1995; Asch, 1952) of use. For simplicity we do not show the social context of users in Figure 1. However, as our ensuing discussion reveals, social context and the manner in which cognition is distributed across a group of users and IT artifacts is an area that should be of increasing focus for cognitive research in IS.

2.2. A Model of Cognition: The Human Information Processor

As noted above, our framework also reflects a cognitive model. We adapt Card et al.'s (1983) model of the human information processor (HIP), which comprises three subsystems. The Perceptual System "carries sensations of the physical world detected by the body's sensory systems into internal representations of the mind by means of integrated sensory systems." (Card et al., 1983, p. 25). The Motor System is where "thought is finally translated into action" (Card et al., 1983, p. 34). Finally, the Cognitive System "connects the inputs from the perceptual system to the right outputs of the motor system" (Card et al., 1983, p. 35). The cognitive system, which appears simple, is more complicated than the other systems, as it involves learning, recall of knowledge, and use of the knowledge to solve a problem (Card et al., 1983). In the HIP model, "internal representations" are central to cognition.

We employ Card et al.'s (1983) model as an organizing framework[3] to identify relevant constructs and concepts in a broader picture of HIP enabled by information systems (see Table 1). In adapting Card et al.'s model for our historical analysis, we focus primarily on the cognitive system, but recognize perception as the source of input to cognition, and use the term "action" instead of motor system, since in an IS setting, the results of cognition are not exclusively motoric.

Notably, the model portrayed in Table 1 extends Card et al.'s (1983) model beyond the individual level of analysis. We follow Hutchins (1990, 1991, 1995) in the use of the term "Distributed Cognition" to describe "the process of thinking as extending beyond the individual … either across members of a group or in concert with objects and tools in the environment" (Gureckis & Goldstone, 2006, p. 293). The term includes but does not require knowledge sharing (and, hence, subsumes "shared cognition"). A distributed cognition view recognizes the collaborative nature of cognitive tasks, whether that collaboration is between a single individual and system or a group of individuals and systems. It may involve communication and sharing of representations between individuals, the development of shared representations (e.g., cognitively: shared mental models, physically: boundary objects). Alternatively, it may involve a coordination of cognitive activities that entails a representation distributed across individuals and systems but in which no one individual or system has the "complete" picture. As our history of cognitive research in IS reveals distributed cognition, with its consideration of knowledge sharing, social context and collaborative task performance is very much "a new foundation" for cognitive research in IS (Hollan, Hutchins, & Kirsch, 2000, p. 174).

---

[2] The role of mental representation in cognition has been the "subject of a lively debate in cognitive science for many years" (Thornton, 2009, p. 1383). Ontologically, the term "mental representation" refers to an observer construct, i.e., an explanatory device rather than something that necessarily exists in the head of an individual. Indeed, we make no claims as to physical implementation of cognitive constructs, as our reference discipline is cognitive psychology rather than neuroscience. Thus, following Neisser (1976, p. 11), we recognize that "perception and cognition are usually not just operations in the head but transactions in the world", and our scope includes aspects of distributed cognition as discussed below.

[3] As a model, its validity against alternatives can be debated. As an organizing framework, the presence of alternatives does not detract from its usefulness in the context of the design and use of information systems. As Card et al. (1983) remark "[i]ts function is synthesis, not discrimination of alternative underlying mechanisms".

Table 1. An Adaption and Extension of the Human Information Processor Model

Individual Level		Beyond the Individual
Individual Level	Beyond the Individual	Individual Level
Perception	Attention	Perception
Cognition	Mental Representations: • Learning and memory (of representations) • Knowledge (as representations) • Reasoning (with representations)	Distributed Cognition: • Knowledge sharing (sharing representations) • Transactive memory (shared representations) • Distributed representations
Action	Judgment Choice Physical act	Collective action Group consensus and choice

2.3. Summary: From Framework to Research Streams

As shown in Figure 1, the concept of representation is central as we explicitly consider developer's mental representations and user's mental representations, in addition to the IS as representation. Thus, we frame our historical analysis of cognitive research in IS into three interconnected streams of research: the development of "representations" (e.g., the cognitive processes of developers), (task level) reasoning with "representations" (i.e., decision support), and the (tool level) interface with "representations" (i.e., human-computer interaction (HCI)). In our review, all three aspects of research on representations appeared in major IS journals. As a matter of scope, our historical analysis has focused primarily on mainstream IS journals, although obviously, we look more broadly in the early days prior to the establishment of IS discipline-specific outlets.

Clearly, good cognitive research in IS should be informed by and build on theory and methods in the foundational cognitive and psychological literature. However our historical analysis reflects the evolution of cognitive research in IS, not cognition research more broadly. We recognize that in parallel with the evolution of IS research, cognitive research also evolves, but we consider this evolution from the perspective of IS research. Our rationale is twofold here. First, this is an IS history written for IS researchers interested in IS phenomena, rather than for cognitive researchers interested in IS as a context for cognitive research. Second, not surprisingly, in our historical investigations we found that the evolution of cognitive research in IS has been primarily driven by the challenges and questions arising in IS, rather than by advances in cognitive theory and method. As such, IS is the source of these questions, whereas the cognitive literature is a source of useful theory and methods for addressing these questions. Although advances in theory and method allow us to address IS questions sometimes in new and different ways (e.g., as in the current interest in Neuro IS), the driving and enduring questions are always about IS. This is as much a matter of pragmatics and socialization: cognitive researchers in IS are on the whole first and foremost IS researchers, not cognitive researchers[4]. Thus, in our ensuing analysis of cognitive research in the three streams, while we identify the cognitive foundations that have informed the IS research, we focus on the evolution of enduring IS questions as the "landmarks" in our navigation through history.

[4] This is evident in even a cursory examination of the publication records of cognitive IS researchers – unsurprisingly, they tend to publish more in IS journals than the underlying reference discipline.

3. Cognition in Software Development

3.1. Scope

The study of cognition during software development has been referred to as software psychology, which stresses the scientific study, modeling, and measurement of human behavior in creating or using software systems (Shneiderman, 1979). Creating a software system is fundamentally a cognitive endeavor; software itself has been described as "pure thought-stuff, infinitely malleable" (Brooks, 1987, p. 12). By implication, software is an outgrowth of human cognition. Consistent with our framework (Figure 1) the studies in this area reflect an interest in the representations of software developers. Early researchers did not usually directly investigate developers' mental representations. Instead, they focused on performance, as the end result of the mental representations implicit to conducting these tasks. Sackman, Erikson, and Grant's (1968) study of debugging performance differences between on-line and off-line environments would likely be considered the first empirical study in this area. Their results highlighted substantial variance between subjects and they, therefore, called for future research focused on "programmer problem-solving". Early studies of software developers addressed this call through a focus on tools and techniques to facilitate software development. These studies typically made limited use of psychological theories, which may have been due to the emerging state of cognitive psychology (the term being coined by Neisser, 1967). Early studies of software developers drew from psychological research in their use of experimental methods. As we note below, as theories of cognition became more developed, cognitive research on software development better incorporated such theories.

3.2. Facilitating Software Development, Implementation, and Maintenance

The earliest studies of cognition during software development sought to understand how to facilitate the cognition of software developers. The investigations focused on how different programming constructs, tools, practices, and so forth enabled the cognition necessary to conduct a programming-related task. The implication was that different techniques were better or worse at supporting the cognitive processes of programmers engaged in tasks such as constructing a program, debugging, maintenance, and comprehension. Studies of software development techniques focus on reasoning with representations (see Table 1) in that development techniques are typically embodied as representations. Consider a Data-Flow Diagram (DFD) or Use-Case Diagram, either would be considered a representation. Even different coding practices, in essence, reflect guidance on how to rely upon the program itself as a representation. Software development can be seen as creating representations (such as DFDs, data models, etc.) and transforming those representations into a representation that is machine executable (i.e., the program code). Software developers rely upon internal mental representations to create and work with these more physical representations (e.g., DFDs, flowcharts).

Studies investigating programming factors, tools, and practices dominated this area from the late 1960s to early 1980s. Sackman et al.'s (1968) study of debugging performance, noted above, was motivated by the then new technological capability of working on-line rather than off-line. Citing Sackman et al. extensively, Weinberg (1971) argued that to understand "the psychology of programming" merited extensive research using a variety of approaches (e.g., introspection, observation, experiments, etc.). As structured programming techniques became prominent, investigations focused on factors such as programming language constructs (e.g., nested conditional constructs) (Sheppard, Curtis, Milliman, & Love, 1979); the role of programming support tools (e.g., testing, flowcharting, and so forth) (Shneiderman, Mayer, McKay, & Heller, 1977); and programming practices (e.g., commenting, variable naming, indenting, and so forth) (Shniederman, 1976). These studies only occasionally assessed cognition directly, such as Sheppard et al.'s (1979) reliance on free recall data to assess program comprehension. Instead, task performance (e.g., programming, debugging, and so forth) was examined to indicate which techniques better enabled programmers to conduct the cognitively complex work of software development. That is, programmers performed better due to improved cognitive processing per the factors manipulated in an experiment such as the presence of code comments, variable naming practices, and the like. These early studies typically lacked a strong theoretical grounding, focusing instead on investigating current software development practices. Hence, the motivation was the on-going evolution of IS practice.

As structured development (SD) expanded to consider modeling techniques for earlier phases of development (Gane & Sarson, 1979), research reflected this evolution (e.g., Vessey & Weber, 1986). In the 1990s, the emergence of object-oriented development (OOD) was a catalyst for inquiry. Several studies focused on comparing object-oriented analysis (OOA) to SD approaches (Agarwal, Sinha, & Tanniru, 1996; Wang, 1996). These studies examined requirements modelling, comparing participants' ability to create representations in the two techniques. Wang (1996) conducted an exploratory study of the cognitive processes used during OOA v. SD via protocol analysis. Agarwal et al. (1996) relied upon the theory of cognitive fit (Vessey & Galletta, 1991), one of the first cognitive theories developed within IS, to argue that OOA and SD approaches are differentially suited to particular application domains. They contrasted performance on the OOA and SD tasks, and their results supported their arguments. Hence, one could not conclude that one approach was always superior with regard to the cognitive task of developing software models.

The standardization of data modeling representations motivated studies similar to those observed in SD. Within the data modeling stream, early studies focused on physical database design (Prietula & March, 1991). To understand how software developers create data models, Srinivasan and Te'eni (1995) analyzed verbal protocol data. They identified several heuristics used to manage the complexity of developing data models.

Ontological analysis, a refinement of philosophy of science arguments (Wand & Weber, 1993, 2002) serves as the theoretical underpinning for many data modeling studies. These studies frequently examined how well predictions based upon ontological arguments are consistent with the cognitive realities of software development – such as developing conceptual models or facilitating understanding. One motivation for ontologies is that if "phenomena are classified correctly according to the theory, humans will be better able to understand and predict the phenomena and thus work more effectively and efficiently with the phenomena" (Wand & Weber, 2004, p. iv). Hence, although one might argue that ontological analysis is not itself a cognitive theory, it has motivated many studies to ascertain if models based upon ontological principles support the cognitive processes of software developers and others who interact with an IS. As data modeling and ontological analysis matured, studies focused on more precise issues such as whether or not attributes and entities should be modeled distinctly based on whether database designers cognitively distinguish them (Weber, 1996) or whether modeling optionality supports designers' cognitions (Bodart, Patel, Sim, & Weber, 2001).

As our understanding of software developers' cognitive processes has evolved, researchers have taken more pro-active approaches. Rather than responding to trends in practice, some suggested practices based upon cognitive theory. Kim, Hahn, and Hahn (2000) modified OO modeling practices to incorporate additional visual cues and contextual information. These modifications were to facilitate perceptual and conceptual integration based on the theory of diagrammatic reasoning (Larkin & Simon, 1987) to help developers integrate information dispersed across multiple diagrams. From verbal protocols, participants relied upon visual cues and contextual information to construct more comprehensive mental representations, resulting in higher levels of problem solving performance.

From these studies, there appears to be a progression from studying programming and physical design to studying tasks that occur earlier in the systems development life cycle (i.e., early research often focused on coding and debugging, later studies considered requirements modeling). However, software design has been under investigated (relative to other software development phases) by the IS community; although many software design studies appear in cognitive science publications (see also Détienne, 2002). Instead, the focus has been on modeling (both data and process), perhaps because these activities are more closely related to the business (problem solving) than the implementation (computing) domain.

Current trends in software development, including test-driven and agile development, have not yet focused on how these techniques facilitate or influence software developers' cognitions (we comment on pair programming later). While we do not advocate following trends per se, it seems appropriate to investigate evolving development techniques to understand how they support developers' cognitive processes and if they do so in ways that are superior to existing techniques. Such research informs managerial questions regarding which changes in development approaches merit the expense and effort to alter development practices. We summarize the above with the following research question:

DEV-RQ1: How can different software development techniques facilitate cognition of developers in the building and maintenance of representations?

3.3. The Role of Specialized Knowledge

Developing software requires extensive knowledge (Sheil, 1981). Many studies have focused on understanding how specialized knowledge enables software developers' cognitive processes while engaged in software development activities. These studies seek to understand developers' mental representations (Figure 1), and their knowledge (Table 1). The focus on the knowledge required to conduct software development tasks has spanned the phases of the software development life cycle. Studies of software developers' specialized knowledge were first conducted by cognitive psychologists, who began to study computer programmers in the late 1970s. In this context, programming was often a mechanism to understand higher-order (problem solving) cognitive processes rather than as a means to improve programmer productivity or software development processes. These studies provided both empirical results and theoretical models that could be applied to IS studies. Other studies of software developers' knowledge structures investigated the cognitive processes of developing a program, which also encompasses aspects of software design (Jeffries, Turner, Polson, & Atwood, 1981; Rist, 1989). These studies focused on how software developers use existing plans or schemas to develop computer programs and sometimes varied expertise in programming and or domain knowledge (Adelson & Soloway, 1985; Solloway & Ehrlich, 1984).

Perhaps the earliest IS study that focused on the role of specialized knowledge was Vitalari's (1985) investigation of systems analysts that identified six major areas of knowledge (core system analysis, high-rated domain, application domain, organization specific, methods, and techniques). These areas were identified from verbal protocols collected while analysts determined requirements from a narrative.

In a similar vein, but at what one might consider the opposite end of the software development life cycle, Nelson, Nadkami, Narayanan, and Ghods (2000) relied upon revealed causal maps developed from individual interviews to identify the knowledge and expertise involved in software operations support (maintenance and related activities). They identified personal competencies, environmental factors, support personnel motivation, IS policies, and support personnel outcomes as important areas of knowledge that these software personnel rely upon to conduct maintenance work. These studies highlight the variety of knowledge required.

Both IS researchers and cognitive psychologists compared expert and novice programmers. Studies of novice-expert differences across domains (i.e., chess, physics, and programming) conclude that experts:

1. Chunk information into familiar patterns and experts' chunks are larger than those of novices (McKeithen, Reitman, Rueter, & Hirtle, 1981).

2. Use a deep structure to organize their knowledge base, whereas novices use surface features to organize information (Adelson, 1981; McKeithen et al., 1981).

3. Work forward to solve a program, whereas novices work backward, setting sub-goals.

Anderson (1995) notes that some studies find that expert programmers deviate from the third conclusion in that they work in a top-down fashion, refining the problem into successive sub-problems. This is considered analogous to working backward. However, expert programmers work in a breadth-first fashion (filling in an entire level). Novices work depth-first, working down to the lowest-level of a particular sub-problem then returning to the higher level. One possible explanation for this distinction between expert programmers and other experts is that programmers frequently apply their skills in a variety of problem domains. As such they may never become expert enough in any particular domain to work forward completely when developing a program. Programmers require extensive interaction with a program to develop a functional representation, which is more consistent with the real-world than the programming domain (Pennington, 1987). The difficulty that programmers experience in developing a mental representation of the problem domain may explain why even expert programmers do not work forward.

To explore the role of expertise in software development, Vessey (1985) developed a theory of debugging expertise via protocol analysis. Vessey found that only novices used a depth-first strategy to debugging. Although both novice and expert debuggers were observed to use a breadth-first approach, expert debuggers also created an overall model of system structure. Schenk, Vitalari, and Davis (1998) investigated novice and expert systems analysts. Novice system analysts made fewer domain specific references. Novices and experts also managed problem-solving hypotheses differently (i.e., they posited a similar number of hypotheses, but novices tested and discarded fewer). Further, novices' approaches differed from those of experts in that novices verbalized fewer goals (goals have been defined as a desired change in state – see Newell & Simon, 1972) and more strategies. However, novices' strategies were top-down but weak; their strategic statements lacked specificity and detail. Experts' strategies were more structured and bottom-up. Their strategic statements contained attainable specific steps, whereas novices' statements were characterized as "lists of actions items" (Schenk et al., 1998, p. 33).

Application domain knowledge, that is, the "real-world" domain that is the source of the problem that is being solved by the programming artifact, has also been extensively studied. Although it is not surprising that knowledge of the application domain is fundamental to systems analysis (Vitalari, 1985), subsequent studies find it relevant to a variety of development tasks. In a study of end-user development, Mackay and Elam (1992) considered how novice and expert levels of healthcare planning knowledge (the application domain) and spreadsheet knowledge (the programming domain) impacted problem solving ability. Their findings were consistent with earlier psychological studies of expertise in that experts utilized deep structure knowledge and a more forward problem-solving approach. They concluded that the "ideal user is one who is very knowledgeable of the functional domain in which the decision of interest resides and who is also very experienced with the software itself" (Mackay & Elam, 1992, p. 168). Similarly, novice analysts were more effective at modeling when they were familiar with the application domain (Vessey & Conger, 1993). They relied upon a general theory of mental models (mental models are a type of internal mental representation, see Gentner & Stevens, 1983; Perrig & Kintsch, 1985) to predict differences in novice analysts' performance. Knowledge of the application domain predicted which of two comprehension processes programmers used to understand a computer program (Shaft & Vessey, 1995). Shaft and Vessey (1995) contrasted theories of computer program comprehension: "top-down" hypothesis-driven (Brooks, 1977) and inferential, "bottom-up" theory (Shneiderman & Mayer, 1979). Subsequent work considered how use of a comprehension process influenced the type of knowledge (programming versus application domain) referenced as well as the effect on comprehension (Shaft & Vessey, 1998). The three IS studies cited above all relied upon verbal protocol analysis. Task performance was also examined in the Mackay and Elam (1992) and Vessey and Conger (1993) studies. Shaft and Vessey (1998) also measured comprehension via responses to questions. These studies indicate that task performance cannot be explained by knowledge alone, but rather requires consideration of cognitive processes to understand how knowledge is applied.

Kim and Lerch (1997) investigated writing a program. Their theoretical perspective relies upon cognitive theories of scientific discovery, modified to consider the representation space. The representation space is a mental model that encodes the programmer's current understanding of the target problem (Letovsky, 1986) and can be considered a type of knowledge structure. Programmers changed representations to lessen the cognitive difficulty (the process) of program generation or testing. Based upon analysis of verbal and program protocols, they conclude that representation (knowledge structure) changes happen suddenly. This study highlights the interplay between cognitive processes and knowledge structures. In addition, the observation that programmers changed their representations to lessen cognitive difficulty is consistent with cognitive fit arguments that a mental representation that does not fit the nature of a task requires the problem solver (e.g., a software developer) to transform his or her mental representation to facilitate problem solving (Vessey & Galletta, 1991).

The theory of cognitive fit also permeates investigations regarding software developers' knowledge. Shaft and Vessey (2006) found performance on a modification task was enhanced when maintainers worked in conditions that created cognitive fit. Fit was operationalized as depending on the nature of the modification task and maintainer knowledge of the application domain. Khatri, Vessey, Ramesh, Clay, and Park (2006) use cognitive fit to establish theoretical differences in the role of application domain knowledge for different types of schema understanding tasks, which was found to be contingent on task type. Both studies conclude that researchers should consider the nature of the application domain and participants' application domain knowledge in subsequent studies.

Similar to our observation with regard to DEV-RQ1, the development of OOA motivated studies of software developers' knowledge. In this context, researchers considered the challenges of shifting from SD to OOA. These studies focused on how knowledge of one development method eased or hindered the transition to the new OOA methodology (Sheetz et al., 1997; Nelson et al., 2009). Sheetz, Irwin, Tegarden, Nelson, and Monarchi (1997) investigate this issue via the elicitation of cognitive maps. Nelson, Armstrong, and Nelson (2009) assess professional developers' knowledge structures at different stages in the transition from SD to OOD to ascertain an understanding of the cognitive shifts necessary during this transition.

Reuse, the application of existing software artifacts in the development of a new system (Irwin, 2002), has been little researched from a cognitive perspective within IS. However, Irwin (2002) studied class reuse via verbal protocol and performance measures. The focus was on how programmers utilized their knowledge to identify and apply an analogous solution. Irwin borrowed from cognitive psychology, applying Gentner's (1983) typology of analogical reasoning.

A different take on understanding the knowledge to support software developers' cognitions appears in recent work investigating how diagrams allow analysts to increase their knowledge of the domain (Burton-Jones & Meso, 2006). Using the "good decomposition model" (GDM) (Wand & Weber, 1990) three versions of UML diagrams were generated with different levels of adherence to the GDM. Unlike earlier studies that operationalized the GDM for systems design, this study applied it to systems analysis. Developers who utilized the UML diagrams that most adhered to GDM demonstrated understanding of and problem solving within the domain. These studies demonstrate a transition that we will observe later: viewing software development artifacts not solely as mechanisms to aid developers, but also as a means to develop the knowledge of those that rely upon them.

The studies described in this section focused on the knowledge that software developers rely upon while engaged in developing software. The earliest studies focused on eliciting and specifying the nature of the knowledge. Later studies, often through the applications of stronger theoretical models, linked specific types of knowledge to their support of different cognitive processes, including theories of program comprehension, analogical reasoning, or learning. We summarize this stream of research with the following research question.

> *DEV-RQ2: How does specialized knowledge enable developers' cognition in creating representations?*

3.4. Knowledge Sharing in Software Development

Although the earliest study we note (Sackman et al., 1968) called for investigations of programming teams, there has been relatively little cognitive research on collaborative issues (e.g., knowledge sharing and others aspects of distributed cognition) in software development. Of interest are the mental representations of both software developers and users (Figure 1). These studies focus more on knowledge (Table 1) than on reasoning with representations. One fundamental collaboration is that between analyst and user to develop system requirements. This is a cognitive issue because at the core of understanding user requirements is a need to share knowledge between the user (or groups of users) and software developer(s). Despite its importance to software development, requirements development has not been subject to extensive research (Browne & Rogich, 2001).

Above, we noted two studies focused on the knowledge of systems analysts (Vitalari, 1985; Schenk et al., 1998). Their focus was on individuals' knowledge rather than the interaction and sharing of knowledge between an analyst and a user. Other studies, however, have sought to understand the analyst-user interaction required to develop requirements (Chakraborty, Sarkar, & Sarker, 2010; Majchrzak, Beath, & Lim, 2005; Marakas & Elam, 1998; Tan, 1994; Zmud, Anthony, & Stair, 1993). Many approaches have been employed to study this process: mock interviews with experimental confederates (Zmud et al., 1993), content analysis of audio and video tapes of interactions between professional systems analysts and clients (Tan, 1994), analysis of questions asked during interviews (Marakas & Elam, 1998), the theory of collaborative elaboration (Majchrzak et al., 2005), and grounded theory development based on interview data (Chakraborty et al., 2010). Despite the importance of requirements determination and ample historical motivation for its investigation, these studies appear somewhat piecemeal, achieving little coherence.

Similar to what we noted in the previous section, a trend seems to be developing to consider the software representation (the IS itself or the documentation artifacts that exist in its context) as a means to shape the knowledge of those who use the system, rather than solely as support for development. Conceptual models can be a means for individuals to increase their knowledge about a business domain (Burton-Jones & Meso, 2008; Shanks, Tansley, Nuredini, Tobin, & Weber, 2008). Burton-Jones and Meso (2008) integrated the Multimedia Theory of Learning (Mayer, 2001) with decomposition quality (per GDM, Wand & Weber, 1990). Having additional representations (i.e., text in addition to a conceptual model) enhanced novices' learning about a domain. Another interesting element of this study is the reliance on multi-media protocol data (screen-cam records, think aloud data, and Webcam videos of participants). Hence, although some studies in this time frame preferred to focus on knowledge structures (e.g., Shaft & Vessey, 2006), newer technology provided additional opportunities for considering process.

Current trends in software development, such as pair programming and the reality of team-based software development, acknowledge the cognitive aspects of these endeavors. A few early investigations of software development teams considered how the mix of cognitive styles represented within a team related to system success (Brittain White, 1984; Kaiser & Bostrom, 1982). These studies relied upon the Myers-Briggs Type Indicator (MBTI), and results indicated that a mix of cognitive styles can be beneficial to success and that different cognitive styles may be more or less beneficial depending upon the phase of the development life cycle (Brittain White & Leifer, 1986). Not all studies have supported the argument that the mix of cognitive styles across a team explains systems development success (Nutt, 1986). Although some recent studies have investigated cognitive style in the context of effective software development teams (Gorla & Lam, 2004) and software developer job fit (Chilton, Hardgrave, & Armstrong, 2005), cognitive style has been not been a consistent focus of software development studies. The equivocal results or the timing of early cognitive style studies (shortly after Huber's (1983) influential paper, which we discuss later) may have contributed to cognitive style being investigated sporadically. However, with regard to software development teams, cognitive style, an individual difference variable, may have limited explanatory power and investigations of other cognitive issues – such as knowledge and process – are likely more informative .

Some early studies of software development teams indicate that knowledge sharing occurs in many areas including the client's requirements, development tools and languages, and so on (Curtis, Krasner, & Iscoe, 1988; Waltz, Elam, & Curtis, 1993). Shared knowledge of the task and of the team appear to be helpful to software development team coordination (Espinosa, Slaughter, Kraut, & Herbsleb, 2007). Knowledge sharing continues to be relevant post-implementation (Santhanam, Seligman, & Kang, 2007). The types of knowledge shared (know-why, know-how, and know-what) differed depending upon the role (user and help-desk personnel) and the direction of the knowledge sharing. There seems to be opportunity for studies exploring knowledge sharing and other aspects of distributed cognition in development.

Despite the studies mentioned above, studies of pairs or SD teams rarely focus on shared cognition. This is perhaps surprising given that "[p]airing developers allows them to share knowledge and form a common understanding of the system and the development tasks" (Dawande, Johar, Kumar, & Mookerjee, 2008, p. 73). Ironically, these authors then take an analytical approach to develop module-developer assignments. Another study investigating pair programming offers a cognitive argument to explain performance on a programming task and finds that pairs' performance was superior to the second-best member of the pair, but not the best member (Balijepally, Mahapatra, Nerur, & Price, 2009). They also examine perceptual measures of satisfaction and confidence in performance but do not examine cognitive processes explicitly. Focusing on performance is consistent with early SD studies (e.g., Shneiderman et al. 1977). It will be interesting to see if studies of pairs and SD teams follow a trajectory similar to studies of individual software developers and begin to investigate cognition more directly, such as through verbal protocol data or a consideration of how the distribution of types of knowledge (e.g., programming, modeling, application domain, and so on) between a pair or across a software development team influences performance. Another issue that seems relevant to software development teams are what and how software development tools and techniques can facilitate shared cognition.

Current work on boundary objects (Zmud & Lim, 2011) provides a theoretical lens for investigating the distributed cognition (especially knowledge sharing) needed to facilitate software development in organizational settings (e.g., Bergman, Lyytinen, & Mark, 2007). Boundary objects are artifacts that are useful in "bridging disparate knowledge structures" (Zmud & Lim, 2011) and include, but are not limited to, diagrams, prototypes, and methodologies. We summarize the above with the following enduring question:

DEV-RQ3: How can software development tools, techniques, and boundary objects facilitate distributed cognition among development teams, users, and managers?

3.5. Summary

Table 2 provides a summary of the evolution of cognitive research in software development. While the efforts in this area began with little reference to theory, cognitive psychology as a reference discipline played an important role in addressing IS questions in software development. The evolution of software development technologies and methodologies creates a nearly endless opportunity for investigation. Early studies focused on changes in the implementation technology, such as the change to on-line programming environments (Sackman et al., 1968); that is, a change in the manner in which software developers interacted with the software itself. It might be argued that some of these early studies pre-date the development of IS as a field, given that *MIS Quarterly* was not established until 1977. From a historical perspective, however, these early software development studies created a foundation and focus if not on cognitive theories (which were emerging), then on the use of the experimental method. These studies were motivated by the underlying notion that software developer performance could be aided or hindered by how well different software development tools and techniques supported the cognitively complex task of software development. As theory evolved, this notion became articulated by the theory of cognitive fit: a representation (either an internal mental representation or a representation as embedded in a development artifact) that fit the task requirements would aid developers by reducing the cognitive load, or cognitive effort, associated with conducting the task. Consideration of knowledge structures was imperative to the development of cognitive fit theory. Not surprisingly, we note many studies, both prior to and since the theory of cognitive fit emerged, investigated software developers' knowledge structures. These studies built on similar studies in cognitive psychology. Perhaps the clearest linkage is investigations of differences between novices and experts. Performance differences between experts and novices are expected, and hence, somewhat uninteresting. Instead these studies investigated the nature of experts' knowledge, its structure and application. The focus on knowledge and how it shapes the ability to conduct a development task continues to be important. We note a recent trend to focus less on individual and more on distributed cognition (i.e., beyond the individual). This focus is consistent with the nature of IS development work, which is nearly always collaborative, and also reflects recent theoretical work in distributed cognition. Software development artifacts applied as boundary objects can facilitate knowledge sharing and structure task environments. The studies in this section illustrate how the contexts and areas of interest evolved with technological developments. Nonetheless, underlying enduring questions are evident, and the full potential for cognitive research in this domain remains to be realized.

Table 2. The Evolution of Cognitive Research in Software Development

Time Period	Context	Theory and Concepts	Areas of Interest	Examples
1970's	Timesharing Operating Systems High level languages		Programming support tools, language constructs and programming practices	On-line v. Off-line debugging (Sackman et al. 1968) Flowcharting (Shneiderman et al., 1977) Programming practices (Sheppard et al., 1979)
1980's	Structured Development	Novice/Expert differences	Logic structuring; Debugging; Design	Expert debugging processes (Vessey, 1985) Decision tables, decision trees, structured English to support programming (Vessey & Weber, 1986)
1990's	Object-oriented development Data modeling	Human information processing; cognitive fit	Requirements elicitation OO v. SD	OO v. SD (Agarwal et al., 1996) Data modeling (Srinivasan & Te'eni, 1995) Transitioning to OOD (Sheetz et al., 1997)
2000's	Conceptual modeling	Cognitive structures; Ontologies Boundary objects	Software maintenance and reuse Team / group cognition; Organizational cognition	OO modeling (Kim et al., 2000) Causal map of maintenance knowledge (Nelson et al., 2000) Reuse (Irwin, 2002) Ontology (Wand & Weber, 2004) Cognitive fit of knowledge structures during maintenance (Shaft & Vessey, 2006) Software development representations to increase developers' knowledge (Burton-Jones & Meso, 2006) User-analyst knowledge sharing (Charkaborty et al., 2010)

4. Cognition and Decision Support

4.1. Scope

From the beginning the term "Decision Support Systems" (DSS) has been interpreted and defined in vastly different ways (Alter, 1977; Bonczek, Holsapple, & Whinston, 1981; Sprague, 1980; Sprague & Carlson, 1982). Like many terms in IS, the definition of a DSS did not arise from some theoretical foundation (such as an understanding of cognition in decision making), but rather from practice (see

Alter, 1977). Consistent across these definitions is the idea that DSS entail acquisition, organization, and use of knowledge representation in the support of semi-structured decision making (Keen & Scott Morton, 1978). Cognition is, thus, central to the nature of DSS, and a major research concern is how well the resulting "man-machine" (user-DSS) combination performs (Blattberg & Hoch, 1990). The essential role of cognition in DSS is evident from the early model-based DSS up to the latest business analytics tools where "comprehensible" outputs, clearly a matter of cognition, are a critical feature for success (Kohavi, Rothleder, & Simoudis, 2002).

While the definition of DSS was informed more by understanding IS practice than psychological theory, the definitions and approaches were essentially consistent with cognitive psychology. Fundamentally, DSS recognize the bounded rationality of users, namely "… the psychological limits of the [human] organism (particularly with respect to computational and predictive ability" (Simon, 1955, p. 101). As such, DSS are intended to compensate for human cognitive limitations, including biases (Tversky & Kahneman, 1974).

While development and user interface issues have been an important part of the DSS literature (Arnott and Pervan, 2005), we address cognition in development and human-computer interaction elsewhere in this paper. In what follows, we explore the enduring IS questions of cognition within decision support rather than provide a review of the DSS literature per se (see Power (2007) for an historical overview or Arnott and Pervan (2005) for a recent critical analysis).

4.2. A Question of Balance in DSS: Models vs. Users

As Simon (1978, p. 479) notes, "humans and computers do not have the same strengths". As far back as the 1950s, it was recognized that they can be complementary (Simon, 1955). Similarly, the question of cognition and judgment in the use of quantitative models has been an interest in the psychology literature (Meehl, 1954) since before the invention of computer-based DSS. This question persists in psychology, as the title of Kleinmuntz's (1990) article reveals: "Why we still use our heads instead of formulas: Toward an Integrative Approach". Even after more than 30 years of consideration, Kleinmuntz believes psychology is still only moving "toward" an approach (Kleinmuntz, 1990). Further, it remains a contentious issue in the practice of psychology (Salzinger, 2005).

The balance issue moves from a psychological question to an IS question when the model ("representation") is embedded in a DSS. Indeed, Alter (1977, p 41) categorized DSS based on the "degree to which the system's outputs could directly determine the decision". Technologically, early DSS were primarily focused on providing access to quantitative models; later activity would move to non-quantitative representations. Four of the seven types of DSS in Alter's (1977) taxonomy have "models" in their name.

From a task perspective, much of the early DSS work focused on planning contexts. For example, the Minnesota Experiments employed simulators for production, procurement, and inventory control (Dickson et al., 1977). Indeed, the production planning context persisted as a stalwart for 30 years (e.g., Davis & Kottemann, 1995; Goodwin, 2005). While this facilitates building a cumulative research tradition (Keen, 1980) by enabling greater comparability across studies, it can also lead to research being viewed as "old hat" because the task context is not novel. The mutual influence of context demands and technology advances in early DSS developments is excellently captured by Keen and Scott Morton (1978):

> *As the technology advanced ... many organizations had also to adjust to an increasingly unpredictable environment. ... organizations began, in a piecemeal fashion, to exploit the opportunities the new technology provided and built small interactive systems and models to help them in their planning* (p. 4).

The notion of interactivity is important here. It is a technological change, but more importantly it changed the nature of how the users cognitively engaged with the IS. Indeed the importance of understanding interactivity persists to the present day, as we shall see later as we look to the future of cognitive research (Section 4).

Interactive use of computer-based models brings challenges that require consideration and analysis of user cognition. Examining this issue, Dickmeyer (1983) contrasted the provision of an interactive model with a static forecast printout and found greater changes in users' policy decisions in the interactive condition. Similarly, McIntyre (1982) found greater volatility in performance for DSS users. Further, users of an interactive DSS performed better but took longer, suggesting that interactivity provided better but less efficient decision making (Benbasat & Schroeder, 1977). In contrast, enhanced interactivity that the DSS provides can be harmful to performance (Kottemann, Davis, & remus, 1994). As Farwell (1984, p. 72) notes, DSS requires: "a balance to be struck between the computer's calculation and logic capabilities and the freewheeling judgment of its human operator". The enduring DSS and cognitive research question evidenced is:

> *DSS-RQ1: How can a balance between user cognition and computer-based representations be achieved so as to maximize performance with a DSS?*

Notably, an appropriate balance does not necessarily mean optimal, as some task contexts may not have a normatively correct choice or judgment, even ex post. The issue of balance has been evident in DSS research for many years (e.g., Blattberg & Hoch, 1990; Davis & Kottemann, 1995; Hoch & Schkade, 1998; Jones & Brown, 2002) and continues to be of interest (e.g., Arnold, Clark, Collier, Leech, & Sutton, 2006; Kayande, De Bruyn, Lilien, Rangaswamy, & van Bruggen, 2009; Mascha & Smedley, 2007). Indeed, some authors have expressed a concern for technology dominance in that a user may defer judgment inappropriately to the technology even in the face of significant economic and legal consequences (Arnold & Sutton, 1998; Hampton, 2005).

4.3. Cognitive Style and DSS

Cognitive theories and concepts were initially introduced into DSS research with Mason and Mitroff's (1973, p. 475) seminal definition of an IS as comprising:

> *at least, a PERSON of a certain PSYCHOLOGICAL TYPE who faces a PROBLEM within some ORGANIZATIONAL CONTEXT for which he needs EVIDENCE to arrive at a solution, where evidence is made available through some MODE of PRESENTATION.*

Not surprisingly, the reference to the importance of the user's psychological type led to the introduction and rapid growth of the experimental method in IS research, notably marked by the "Minnesota Experiments" (Dickson et al., 1977). Mason and Mitroff (1973) defined psychological type as cognitive style, and hence, cognitive style became a central variable in DSS experiments conducted at the time (Benbasat & Taylor, 1978).The focus on "type" played into a contingency view of cognitive attributes of users, which no doubt contributed to its popularity, as it was the heyday of contingency research in the organizational studies arena.

Cognitive style refers to an individual's approach to information acquisition, analysis, evaluation, and interpretation (Sage, 1981). It is most commonly conceptualized as two contrasting extremes: analytical versus heuristic/intuitive (Huysmans, 1970), or via a Myers-Briggs Type Indicator (MBTI) (for a recent IS example, see Barkhi, 2002), or through a direct reference to physiology of the brain and hemispherical specialization (e.g., Robey & Taggart, 1982).

In the 1970s and early '80s there was a burgeoning interest in cognitive style research (e.g., Bariff & Lusk, 1977; Benbasat & Taylor, 1978; Henderson & Nutt, 1980; Lusk & Kersnick, 1979). However, the proliferation of interest in cognitive style proved problematic. Most notable was Huber's (1983) well-known critique. In his pejoratively titled paper, "Cognitive Style as a Basis for MIS and DSS Designs: Much ado about nothing?", Huber argued that cognitive style could not form the basis for guidelines for DSS design. Further, future cognitive style research was unlikely to yield guidelines for design. To support these claims Huber cited a multitude of factors, including inconsistent empirical results and measures, lack of theory, but most fundamentally that “we do not know if DSS designs should (1) conform to the user’s cognitive style or (2) complement [it]” (Huber, 1983, p. 571).

In hindsight, we can see that Huber’s (1983) critique recognized that the cognitive style research failed to address the enduring questions of the field. Specifically, the research could not provide guidance as to the balance between computer-based models and managerial cognition (DSS-RQ1) nor could it adequately address the design imperative in the other cognitive research questions that emerged (see DSS-RQ2 and DSS-RQ3 below).

From a historical perspective, we can observe that the rise and fall of cognitive style research was more fundamentally problematic than a failure to adequately address the design imperative. In its rise, cognitive style research in IS became de facto synonymous with cognitive research in IS in general. Thus, Huber’s (1983) treatise that ended cognitive style research also curtailed interest in cognitive research in IS more broadly. The fundamental problem was that cognitive style was an individual difference variable, “it categorizes individual habits and strategies at a fairly broad level and essentially views problem solving behavior as a personality variable” (Keen & Scott Morton, 1978, p. 74). Yet, cognitive psychology is not about personality, it is about the “activity of knowing: the acquisition, organization and use of knowledge” (Neisser, 1976, p. 1). Huber’s (1983) article curtailed future cognitive style research, but unfortunately also impacted all cognitive research in IS, including research that was not focused on individual personality differences. Indeed, it possibly contributed to the relative lack of interest in cognition in the Group Decision Support Systems (GDSS) literature, which was in its infancy at the time of Huber’s critique (although, this may have been due to the absence of adequate theories of group cognition at the time, which we discuss below).

4.4. Toward a Process View

Ramaprasad (1987) attempted to re-direct the field away from cognitive style to a focus on cognition and cognitive processes. More than just a response to Huber’s critique, the call for a focus on cognitive processes was consistent with the broader psychological literature; for example, Payne’s (1976, p. 368) lament in the psychology literature that “most research on decision behavior has focused on data which reflect only the end product of decision processes”. There was a need for a richer view of user cognition and behavior with DSS, and the enduring question emerged:

> *DSS-RQ2: How do DSS design characteristics impact user cognitive processes and performance?*

Pracht and Courtney (1988) sought to direct research toward issues of cognitive ability and cognitive processes in understanding performance with DSS. Their work is particularly notable in that it highlights a distinction between two phases of decision making: problem formulation or structuring and problem analysis (see also Paradice & Courtney, 1986). Indeed, the notion of decision support occurring by structuring the decision problem is one that persists through to the present (e.g., Mennecke, Crossland, & Killingsworth, 2000; Phillips-Wren, Mora, Gorgionne, & Gupta, 2009), perhaps no more so than in the twin concepts of Decisional Guidance and System Restrictiveness, which Silver defines as:

> *Decisional Guidance: the degree to which and the manner in which a Decision Support System guides its users in constructing and executing decision-making processes, by assisting them in choosing and using its operators. (Silver, 1990, p. 57)*

> *System restrictiveness: the degree to which and the manner in which a DSS restricts its users’ decision-making processes to a particular subset of all possible processes. (Silver, 1988, p. 52)*

From the perspective of our DSS research questions, guidance and restrictiveness are key design characteristics that may affect user cognition (DSS-RQ2) and, as a consequence, the appropriate balance between user and models (DSS—RQ1).

Empirically, Decisional Guidance has been found to influence user decision processes and key outcome variables including efficiency, decision quality, user learning, and satisfaction (e.g., Jiang & Klein, 2000; Parikh, Fazlollahi, & Verma, 2001; Singh, 1998). However, the Decisional Guidance literature has been plagued somewhat by the variety of different forms that guidance can take theoretically, and the even greater variety in how those forms are operationalized (Parikh et al., 2001). For example, in an Expert Systems (ES) context, Davern and Parkes (2010) find that the taxonomy of types of guidance appears to be incommensurable with a well-established taxonomy of the forms of explanation mechanism in ES (Gregor & Benbasat, 1999). Regardless, explanation mechanisms are a prime practical example of deliberate decisional guidance. In parallel there is some work in IS (e.g., Bui & Lobbecke, 1996; Kayande et al., 2009; Te'eni, 1991), and a large body of work in psychology examining the role of feedback in judgment and decision making (see Cooksey, 1996, for a review). How this reconciles with work on Decisional Guidance remains unclear.

Empirically users seem to prefer less restrictive DSS (Wang & Benbasat, 2009), although the performance impacts are less clear. Goodwin, Fildes, Lawrence, & Stephens, (2011) find no benefit for restrictiveness in a context where the expectation was that restrictiveness would be helpful. Davern and Kamis (2010) find users of a more restrictive DSS sometimes outperforming those relying upon a less restrictive DSS, although the expectation was that restrictiveness would hurt performance. There is clearly scope for more work exploring user cognition and system restrictiveness.

The increased concern for process in DSS research is also evident in applications of the effort-accuracy framework (see Payne, Bettman, & Johnson, 1993, for a detailed review of the framework). Whereas decisional guidance and system restrictiveness are introduced as design attributes of DSS that impact cognitive processes, the effort/accuracy work started from a psychological theory and sought design implications. The effort-accuracy paradigm in behavioral decision theory holds that individuals use a variety of decision strategies (Johnson & Payne, 1985) and that the chosen strategy is a trade-off between the cognitive effort required to execute it and the expected accuracy obtainable from application of the strategy (Payne et al., 1993). In introducing the theory to DSS research, Todd and Benbasat (1991, p. 90) note:

> *For DSS researchers, the key message to be taken from this work is that decision makers are not solely concerned with decision quality, a factor which has been the main focus of most DSS research to date. Effort considerations can influence the choice of a decision making strategy and as a result decision quality.*

Todd and Benbasat's (1991) comment reflects the shift from DSS-RQ1, where the focus was primarily on decision quality, to an expanded view considering the impact of DSS design characteristics on user decision processes (i.e., the chosen decision making strategy) and task performance (i.e., decision quality), per DSS-RQ2. More pointedly, Todd and Benbasat (1999, p. 356) illustrate the centrality of DSS-RQ1 and DSS-RQ2 to the effort accuracy work, as they seek to address two issues:

> *1) the conditions under which DSS use leads to effectiveness or efficiency outcomes, and 2) the means by which the DSS designer could influence the behavior of the user, and what the extent of this intervention should be.*

Many papers have employed the effort-accuracy paradigm in their theoretical foundations (Davern & Kamis, 2010; Benbasat & Todd, 1996; Song, Jones, & Gudigantala, 2007; Todd & Benbasat, 1991, 1992, 1994a, 1994b; Wang & Benbasat, 2009). Their results suggest that DSS enhance performance by reducing the effort required to employ a more accurate or effective decision-making strategy. There are interesting subtleties here, such as the finding that effort/accuracy trade-offs can persist in the face of incentives (Todd & Benbasat, 1999), and that the trade-off may operate with a decision quality threshold (Chu & Spires, 2000).

The relative success of this stream of research can be attributed to multiple factors. The existence of a strong theoretical basis outside IS and strong methodological basis presented within the IS literature was certainly contributory. For example, Todd and Benbasat (1987) catalog five different methods for "process tracing" including concurrent verbal protocols for which the article is perhaps best known. There has also been methodological sophistication in the measurement of accuracy, for example using Data Envelopment Analysis (Davern & Kamis, 2010). More pragmatically, the rise of e-commerce and the interest in supporting consumer decision making (for recent examples see Davern & Kamis, 2010; Tan, Teo, & Benbasat, 2010; Wang & Benbasat, 2009) added to the longevity of effort/accuracy work, without which the experimental tasks employed in the research would appear to be of far less topical significance.

While the tenets of a cognitive cost/benefit framework continue to be referenced in the current literature, its role has become less central. For example, Tan et al. (2010) adopt a "resource-matching approach" (cognitive resources matching task demands – an issue of cognitive fit, albeit somewhat different to Vessey's concept described above), whereas Davern and Kamis (2010) explore the role of domain knowledge in DSS; but both studies to varying degrees build off the effort/accuracy work. More broadly, the framework has also informed other conceptual developments that have had some bearing on DSS research, notably the work on Cognitive Fit (Vessey, 1991; Vessey & Galletta, 1991) and Task Technology Fit (Goodhue, 1995; Goodhue & Thompson, 1995). In the case of the task-technology fit, studies have generally not been concerned with issues of cognition (but see Davern, 1996, 2007). While this is clearly not the case for cognitive fit, the application contexts have been more around HCI and software development than DSS.

4.5. Sharing Knowledge Beyond the Individual

In parallel to developments with more traditional DSS, the field expanded in the 1980s to include expert systems (ES), a technology itself coming from an area of cognitive science (in the broad sense of the term) – Artificial Intelligence. Despite this connection to cognition, much of the early ES research focused on issues of implementation (Duchessi & O'Keefe, 1992, 1995; Yoon, Guimaraes, & O'Neal, 1995), although, see Ye and Johnson (1995) for an example of cognitive ES research. Later research became more concerned with understanding cognition to influence ES design. For example Arnold et al. (2006) look at different explanation systems and how experts versus novices rely on system recommendations as a result. Gregor and Benbasat (1999, p. 497) observe that "explanations, when suitably designed, have been shown to improve performance and learning and result in more positive user perceptions of a system" (see also Einning & Dorr, 1991). Notably, Gregor and Benbasat (1999) identify cognitive effort as critical in determining what is "suitably designed" (see also Dhawihal & Benbasat, 1996; Mao & Benbasat, 2000). Thus, DSS-RQ2 has persisted into the ES research, albeit in a somewhat different manner.

In the last decade interest in ES has waned. In part this is reflective of the technology becoming embedded in other systems (Metaxiotis & Psarras, 2003). While ES research in IS was "rebadged" with papers preferring the term "knowledge based systems" rather than ES (see for example, Arnold et al., 2006; Gregor, 2001; Gregor & Benbasat, 1999; Hendriks & Vrien, 1999; Mao & Benbasat, 2000; Nah & Benbasat, 2004). The rise of the term "knowledge based system" runs almost parallel with the recognition of knowledge management as a strategically important area of business practice (Davenport & Prusak, 1998; Nonaka, 1991). We discuss knowledge management and knowledge management systems further below.

Concurrently with the rise of ES, was the development of GDSS. In 1987, DeSanctis and Gallupe's foundational work on GDSS was published, citing both "technological advancements" and a "post-industrial environment characterized by greater knowledge, complexity and turbulence" (p. 589) as influences. However, the GDSS literature had less interest in cognition, with a greater focus on interpersonal communication: "although support of cognitive processes of individual group members may be included in a GDSS, the primary aim of the group component of the system must be to alter the structure of interpersonal exchange" (p. 592). DeSanctis and Gallupe's notions of Level 2 and Level 3 GDSS are pertinent here. Level 2 GDSS "provide decision modeling and group decision techniques aimed at reducing uncertainty and "noise" that occur in the group's decision process" (p. 593). Level 3 GDSS incorporate "machine-induced group communication patterns and can include expert advice in the selecting and arrange of rules to be applied" (p. 594). These concepts at a group level are akin to the individual-level role of DSS in structuring tasks (Paradice & Courtney, 1986; Pracht & Courtney, 1988;) and studies of decisional guidance and system restrictiveness.

A decade later, Nunamaker, Briggs, Mittleman, Vogel, & Balthazard (1997, p. 163) provided a review of "lessons from a dozen years of group support systems research", which revealed much of the lessons learnt were more concerned with social interaction: for example, leadership style and appropriate voting mechanisms rather than a strong notion of group cognition. This is not surprising as the requisite theories of cognition had not yet been developed. The concept of "distributed cognition", in which cognition is viewed beyond the bounds of a single agent, did not gain recognition in the cognitive science literature until the seminal work of Hutchins (1995, 1991, 1990), nor had Weick's (1995) or Weick and Roberts' (1993) notions of collective mind and organizational sensemaking become broadly noticed until GDSS research was well under way. Consequently, the views of cognition in GDSS tended to focus on the individual-level cognitive phenomena (e.g., attention and memory (Potter & Balthazard, 2004) and information overload (Grise & Gallupe, 2000).

A key goal of both GDSS and ES developments was sharing knowledge, an aspect of distributed cognition. With ES, this sharing was through the "transfer of expertise" via the system, whereas GDSS often served as a vehicle for sharing knowledge between the individuals and the group as a whole, with a view to enhancing problem solving. The enduring IS question evidenced here is:

> *DSS-RQ3: How can DSS enable distributed cognition between individuals and systems, and among groups and organizations?*

With the question framed in this manner, the issues endure beyond ES and GDSS technologies to knowledge management and business intelligence systems we discuss further below. While GDSS and ES researchers may not have framed their research in the terms of DSS-RQ3, the question is clearly consistent with the objectives of their research. This is evident both in the concern for determining appropriate design features, and the desire to improve individual, group, or organizational problem solving. More broadly, the question was well recognized by Boland, Tenkasi, & Te'eni (1994, p. 456), who argued that:

> *much of the effort to design information technology to support cognition in organizations has not addressed its distributed quality. Such systems have tended to focus either on the individual as an isolated decision maker, or on the group as a producer of a decision or policy statement in common. In distributed cognition, by contrast, the group is a set of autonomous agents who act independently yet recognize that they have interdependencies.*

Boland et al. place somewhat different demands on IT here. It is not about aiding an individual decision maker as in traditional DSS or ES contexts. Nor is it about aiding a group as the decision making agent. Rather, it reflects the interdependencies among decision makers, and that the key is supporting the cognition of individual decision makers who are aware of their need to interact with others to achieve task objectives. Notably, we can see this as IS re-defining again the nature of the term interaction, in a richer and more interesting manner: IT-enabled interactivity amongst decision makers. Similarly, we see expanded notions of fit arising in the group context (e.g., Dennis, Wixom, & Vandenberg, 2001; Zigurs & Buckland, 1998). Progress in cognitive research in IS in the group and distributed context will entail connecting cognitive research with the broader social context to understand the use of IT for information sharing and knowledge sharing (e.g., consider Miranda & Saunders' (2003) analysis of the social construction of meaning in a group decision making context).

4.6. Toward the Future

Where, then, has decision support as an area moved in the 21st century? In addition to the consumer-oriented DSS noted above, the two broad classes of technologies that seek to support decision making and problem solving are knowledge management systems (KMS) and business intelligence/business analytics tools (including OLAP).

In a review of the knowledge management systems literature, Alavi and Leidner (2001) identify five broad research questions, looking in turn at knowledge creation, storage and retrieval, transfer and application. This is consistent with the earlier noted definition of cognition as "the activity of knowing: the acquisition, organization and use of knowledge" and involving both processes and knowledge structures. It seems that a KMS is essentially an IT tool to support cognition, particularly distributed cognition in an organization. The IT related questions Alavi and Leidner (2001) identify are thus consistent with DSS-RQ3 above, although all too often the theory and analysis is predominantly social or organizational rather than both cognitive and social. Indeed, Weick and Roberts (1993, p. 368) lament regarding organizational cognition more broadly still seems apt, if not for organizational studies then at least for IS studies:

> *Discussions of collective mental processes have been rare, despite the fact that people claim to be studying "social" cognition. The preoccupation with individual cognition has left organizational theorists ill-equipped to do much more with the so-called cognitive revolution than apply it to organizational concerns, one brain at a time.*

Business intelligence (BI) (often used synonymously with business analytics, e.g., Wixom & Ariyachandra, 2011) refers to a broad range of technologies and tools for data analysis and reporting that inform and enhance the decision making of users at a variety of levels in an organization (Elbashir, Collier, & Davern, 2008). While BI has become well established in practice, there is surprisingly little behavioral, let alone cognitive, research examining these tools. Indeed, Arnott and Pervan (2005) note as a serious concern the relatively low number of BI papers in the DSS research literature, suggesting it may be a critical crisis of relevance. Similarly, a survey for the BI Congress received responses such as "[f]or academics, BI is a fad at worst or a fashion at best. In either case, it is not new and therefore does not garner the research focus of many" and "[t]he theoretical fundamentals in BI are very weak" (Wixom & Ariyachandra, 2011, p. 6). Understanding cognition with BI tools provides a possible basis for more theoretically grounded BI research, and may assist in building a cumulative tradition, rather than providing commentary on the latest fad or fashion. Moreover, the plight of BI research highlights the importance of conceptual and theoretical developments in decision support research, rather than simply following changing task contexts and advances in technology.

4.7. Summary

Cognition has obviously played an important role in decision support studies in the IS literature. Table 3 provides a summary of the evolution of cognitive research in the decision support domain. While the research literature has adapted to technological advancement and the changes in context of use and areas of interest, there is also evidence of gaps in the literature, and opportunities for future research. As noted above (and, hence, not shown in Table 3), there is relatively little cognitive research examining BI systems, despite their economic significance and widespread use. Similarly, there have been missed opportunities in regard to cognitive research extending beyond the individual-level analysis. The relative paucity of cognitive perspectives on GDSS and knowledge management systems is of note, despite the clear identification of the need to study IT-enabled distributed cognition (particularly, knowledge sharing) (e.g., Boland et al., 1994). On a more positive note, we have seen the development within IS (rather than imported from psychology), of design-relevant concepts like decisional guidance and system restrictiveness in which cognition is central. We have seen powerful explanatory constructs such as cognitive fit emerge. In short, opportunities are abundant for future cognitive research in decision support, but the directions are challenging. Studying IT-enabled distributed cognition, for instance, may require moving outside the laboratory. Even within the laboratory, constructs like decisional guidance, while useful, have been difficult to operationalize consistently.

Table 3. The Evolution of Cognitive Research in Decision Support

Time Period	Context	Theory and Concepts	Areas of Interest	Examples
1970s to early 1980s	Model-based DSS (e.g., for production planning, forecasting, etc.)	Psychological Type DSS characteristics	Effectiveness and efficiency under variations in information load, time pressure, system interactivity, e.g., with visual display terminals, timesharing systems.	Information overload in production planning (Chervany & Dickson , 1974); "The Minnesota Experiments" (Dickson et al., 1977) Interactive DSS in policy decision making (Dickmeyer, 1983)
Late 1980s to 1990s	Choice Tasks Supporting groups and teams DSS as an agent of change Expert Systems	Cognitive Processes Effort/Accuracy Levels 1, 2, 3 GDSS Decisional Guidance, System Restrictiveness Cognitive Learning Theory	Decision Process and Decision Quality Group Decision Making Electronic brainstorming Experiential learning ES explanations	Process Tracing (Todd & Benbasat, 1987) Preferential choice (Todd & Benbasat, 1991, 1992, 1999) Task/Member proximity (DeSanctis & Gallupe, 1987) Production Blocking (Gallupe, Cooper, Grise, & Bastianutti, 1994) Decision Support Theory (Silver, 1991) Learning by Novices (Eining & Dorr, 1991) Acceptance and Types of Explanations (Ye & Johnson, 1995)
2000s	Consumer DSS	Effort/Accuracy	Purchase intention	Acceptance and e-commerce (Kafouris, 2002) Multi-attribute decisions in e-commerce (Kamis, Koufrais, & Stern, 2008)

5. Cognition in HCI

5.1. Scope

Human computer interaction (HCI), as studied within the field of IS, has traditionally focused on processes and outcomes of users interacting with computers to accomplish organizational tasks. One of the first research groups to embrace HCI was a series of symposia, beginning in 1986, called Human Factors in MIS, initiated by Jane Carey, which looked at human aspects of both users and developers. Furthermore, human factors included physical, cognitive, and affective aspects of human behavior. In this section, we look at cognitive aspects of users interacting with computers. Moreover, as suggested by Figure 1, we concentrate on the user's interaction with the computer (the tool) that serves to advance some higher order task such as decision making (discussed in the previous section).

Human interaction with computers is fundamentally a human activity of communication based on a flow of information, i.e., commands and messages, from the user to the computer and back to the user, for generating, using, and manipulating representations. By definition, therefore, it involves cognition as indicated in Figure 1. It is no surprise, therefore, that cognition has played a paramount role in understanding, modeling, and designing HCI.

5.2. Cognition and Performance

Early HCI research in the 1970s experimented with different displays, both in print and on screen, to explore their impact on performance. Frameworks, such as that in Mason and Mitroff (1973), offered a list of relevant parameters, some of which were of interest to HCI, like psychological type and presentation mode. IS researchers explored the impact of various combinations of these variables, e.g., the fit between certain cognitive styles and presentation format (tables versus graphs). The inconclusive results and paucity of theoretical insights was frustrating and led to a near stop of research on cognitive styles, as noted earlier. The emergence of cognitive models, such as the popular keystroke model (Card et al., 1983), moved the research to theory-based experimentation with different designs. As systems became more interactive, research concentrated on online work rather than off-line, and screen design rather than report designs. Interactive work stressed the cognitive difficulties resulting from the user's limited memory and processing capacity.

Cognitive models were first applied to highly structured tasks (e.g., editing operations) and only later adapted to more complex tasks (e.g., working with a spreadsheet). Models of complexity (Berlyne, 1971) were applied to decision-making tasks. HCI issues such as the use of windows to support complex tasks relied on complexity theory (Te'eni, 1989). Moreover, cognitive research looked into the process of interaction rather than only impact (Todd & Benbasat, 1987). This move to process was matched with the use of appropriate research methods such as protocol analysis. This mirrors a similar pattern we observed in the software development and decision support streams.

The design implications were based on the assumption that designers should identify cognitive difficulties in performing the task and attempt to minimize them by careful design. In some cases, tables of “man better than machine” were used to allocate tasks between man and machine according to whether human cognition was better or worse than computer intelligence. But these characterizations were very general. Researchers proposed design methodologies that could systematically represent a particular task and identify the required cognitive resources as the basis for determining effective designs (Zachary, 1988). Research that considered how cognitive resources were used and how their limitations affected performance was, therefore, essential. The research here in HCI clearly is related to the work discussed earlier in the context of DSS-RQ1, the difference here is the focus of interest in the interface and presentation of the IS rather than the underlying representation or decision model itself.

The rapid technological advancements, reflected in higher interactivity and accessibility and, importantly, the evolving theoretical basis, underscored the enduring research question despite the changes in the type of task studied:

HCI-RQ1: How do IT interfaces impact cognition and performance?

Performance, as a function of cognition, was operationalized by the accuracy and efficiency of the task achievement. Tasks ranged from simple editing functions to complex planning and problem solving, but the general thesis was that limited cognitive resources produce better or worse performance according to the particular design. The realization that different designs produced different results for different tasks led to the idea of cognitive fit between computer presentation and task representation (Vessey & Galletta, 1991). Recall that mental representations (sometimes called cognitive models) explained how fit can be achieved. Here too, earlier studies compared the performance of different combinations of presentation and task and only later attempted to uncover the process of attaining fit and using it effectively. And again, to capture cognitive process, appropriate research techniques such as visual protocols were needed (Kennedy, Te’eni, & Treleavan, 1998). Using such protocols, the researchers were able to break the user’s behavior into EIPs (elementary information processes) and calculate the optimal fit by minimizing cognitive effort.

Cognitive fit implied customization of designs to specific tasks. When taken further to include not only adaptation to task but also to the user, customization became personalization. The former research question expanded to underscore customization, in general, and personalization, in particular.

Technological advances made it easy to customize an interface to particular tasks (e.g., change the menu according to expected activity) and to particular users (e.g., adapt the online banking menu to a particular customer). Moreover, people have grown to expect such personalization in business, education and health. Thus, technology advancements and users' expectations and norms of use, one feeding into the other, underscored the need to understand and design customization:

> *HCI-RQ2: How does customizing/personalizing IT interfaces impact cognition and performance?*

The above two research questions have extended in several directions. First, new technologies, often coupled with new social expectations, have introduced new aspects of IT interfaces that "beg" to be studied. For example, HCI researchers have studied mobile IT and context-aware systems. In particular, highly interactive and highly vivid systems have drawn research on visualization and animation (e.g., Hess, Fuller, & Campbell, 2009; Zhang, 2000). Not only has the same research question been sustained, but some of the main theoretical models have been retained and extended, e.g., visualization based on cognitive fit (Goswami, Chan, & Kim, 2008). Second, the context of HCI has expanded. The context has evolved from a near confinement to tasks at the work place to include also tasks in the market place, particularly online consuming, which has accelerated since the 1990s, and more recently, to tasks in the social arena as well (Babu, Singh, & Ganesh, 2010; Hess et al., 2009).

5.3. Beyond the Individual

Research has progressed over the years from studying the individual interacting with a computer to examining several individuals working with computers in teams, to considering people communicating and working in communities. This move required new theory for understanding cognition in interpersonal situations. Some of the concepts on cognition could be used in computer-mediated interaction between individuals, while others had to be developed anew. A primary difference between an individual working alone with a computer and a dyad or group of individuals interacting via computers is the need for communicating. Earlier theories explaining the fit between communication and media such as media richness theory (Daft & Lengel, 1986) and social presence theory (Short, Williams, & Christie, 1976) addressed cognition only marginally. Recent theories delve deeper into cognition. For example, similar to the notion of cognitive complexity in individual work, the notion of communication complexity explains communication outcomes (Te'eni, 2001). Moreover, as most computer-mediated communication occurred between people situated in remote locations, the impact of distance on cognition and ways IT can overcome the negative effects became an important research area (Olson & Olson, 2000).

But there was another research issue that arose in communication and collaboration between individuals, namely, the question of how to support interaction between individuals with different perspectives. Research in HCI tackled this issue at two levels: 1) differences in perspectives that reflect differences in knowledge and attitude (see our earlier discussion of distributed cognition in DSS) and 2) differences in perspectives that reflect different physical views; for example, different displays and room settings in different locations or different groups of people in different locations. The latter aspect of perspectives required research on IT interfaces that bridged the gap between the different views by, say, showing multiple views on different screens or changing views according to who is speaking. It also highlighted the need for awareness of not only what is happening on the screen but what is happening with the people with whom the user is interacting.

Here again, technological advances introduced new challenges. New communication technologies were different from e-mail. Wikis, blogging, and micro-blogging required new cognitive skills such as attention to huge amounts of information, high interactivity coupled with very short messages, and awareness of multiple others simultaneously, which underscored the importance of communication complexity, distance, perspectives, and awareness. Parallel to the technology advancements were the changing practices and norms of work. Increasingly, it has become popular and expected to work in teams, often cross-functional and international teams; to work in flatter, knowledge-intensive organizations; and to work in networks across organizations such as in communities of practice.

Thus, again, the combination of technology and norms has underscored the quest for going beyond the individual to understand distributed cognition:

HCI-RQ3: How do IT interfaces impact distributed cognition between individuals and among groups, organizations, and communities?

5.4. Beyond Performance, Beyond Cognition

Cognitive research in HCI has, in part, moved to consider impacts beyond performance such as attitudes. Moreover, the last decade has seen a strong move to study cognition in conjunction with emotion, a move that strengthened in parallel to the continued trends mentioned above, namely advances in IT, the expanding context of HCI, and more interpersonal and community interaction with computers. As we have seen throughout our historical analysis, the enduring questions emerge more from the evolution of IS phenomena than from cognitive theory, although advances in cognitive research assist in providing an ever deepening understanding of IS phenomena.

At the same time the first two research questions around cognition and performance were studied, HCI researchers engaged in the study of attitudes. The Technology Acceptance Model (TAM) was the most popular model used in IS (Davis, 1989), and was derived from the Theory of Reasoned Action (TRA) (Ajzen & Fishbein, 1980). While attitudes are clearly influenced by cognition -- perceptions and beliefs – the link between cognition and HCI variables was not articulated in this stream of research. More recent treatments of TAM2 attempt to tie cognition into the model more closely (Venkatesh & Davis, 2000). Other research has sought the playfulness or "cognitive spontaneity" of IT use and its impacts on attitudes (e.g., Webster & Martocchio, 1992). Current research refers to cognitive aspects of, say, knowledge processing, but does not uncover cognition to the extent it can explain the impact of HCI on attitudes (e.g., Zhang, Scialdone, & Carey, 2009). We believe future research may extend and intensify this direction.

Furthermore, the number of HCI studies on emotion more than quadrupled in the last decade versus the 1990s (this is the trend in the HCI literature outside IS too). More importantly, a growing number of studies combine cognition with emotion (e.g., Cyr, Head, Larios, & Pan, 2009; Deng & Poole, 2010). As in the HCI field, in general, outside IS, researchers include emotion to explain users' attitudes and choices in settings such as browsing and shopping in virtual stores. But emotion is strongly linked to cognition and, hence, we increasingly find more studies of both. Moreover, cognition and emotion are tied to attitudes, so we can expect HCI models that link cognition, affective emotions, and attitudes to explain behavior (e.g., Van den Heijden, 2004).

At least in part, the growing interest in emotions stems from the expanding context of HCI, which is related to the dramatic increase in accessibility to the Internet and more recently to mobiles and smart phones. If people do not take to a particular website, they go to another; if consumers do not like to shop in one e-store, they turn to another. Hence emotions matter, in addition to cognition, and emotive designs are becoming the norm. Technological advancements for affective computing are evolving gradually (see, for example, the MIT media lab), but much can be done with current technology to improve the aesthetics and positive reactions to the interface. Furthermore, advances in research that combine cognition and emotion make it possible to extend cognitive HCI research to include emotions, too:

HCI-RQ4: How do IT interfaces affect cognition and emotion and, thereby, attitudes and performance?

5.5. Expanding to All Users, Extending Strengths

The expansion of design goals from performance to satisfaction and to quality of life has paralleled an expansion to wider user populations, including the less gifted and less able (the principle of social responsibility). Cognition, of course, plays an essential role in understanding how to provide for all users. For example, designing for the blind requires creative designs that support cognition in the absence of sight (Babu et al., 2010). As IT becomes ubiquitous, we are expected to ensure digital gaps do not worsen the position of the less advantaged, either interacting individually with computers or interacting with others through the Web and other systems.

The HCI perspective of cognition elaborated above builds on models of human information-processing that assume scarce cognitive resources. Different periods have emphasized different cognitive limitations. Human memory and information processing were especially relevant in earlier periods, while attention, awareness, and understanding vast amounts of information seem to be especially relevant currently. Nevertheless, the dominant perspective remains one of scarce cognitive resources, and, when the resources are over-taxed, cognition is impaired. It follows that IT should be designed to mitigate the demands on cognitive resources and the consequent difficulties; for example, reducing memory load or overcoming the impact of distance on poor awareness (Olson & Olson, 2000). Positive design, on the other hand, seeks to leverage the strengths of humans (Carroll, 2010). For example, in some cases, generativity (the ability to originate, produce, or procreate) and not performance should be the preferred outcome (Avital & Te'eni, 2009): IT should be designed to enhance human creativity. To that end, designs that are evocative, adaptive, and open-ended, and the characteristics of IT that support these ends are preferred. For instance, two characteristics from an HCI-specific perspective are visualization and integration. Visualization enables users to see multiple dimensions, such as visual representations, providing the ability to see an object from multiple perspectives and to search for new insightful points of view. Similarly, integration enables linking and aligning interdependent domains, objects, or processes to provide the ability to overlay or merge views and to promote system-wide boundary crossing. This change in the research perspective, while not always applicable to systems supporting routine tasks, seems especially promising for dynamic, agile, diverse, and multi-national organizations.

> *HCI-RQ5: How do IT interfaces enhance cognition for diverse individuals, groups, and communities*?

5.6. Summary

Our central research questions have been shaped to a large extent by the combination of advances in IT, both in hardware and software, and changes in the norms and expectations of designing and using IT. The changes in norms and expectations are also reflected by the design goals of new IT, the new architecture, functionality, and interfaces of the systems. Earlier models of HCI were designed for performance in structured and repetitive tasks. Speed and error rate were the main performance parameters studied. Cognitive models (e.g., GOMS in Card et al.'s, 1983, keystroke model) were developed to predict speed and errors in simplified models of memory and processors. More advanced models were needed to cope with less structured tasks such as decision making, adapting performance to account for decision processes and outcomes (e.g., cost-benefit models and cognitive fit models). In the past 10 years, positive emotions in human computer interaction and positive attitudes toward systems have grown to be expected; cognition must be integrated with emotion to understand humans interacting to achieve these expanded goals. And more recently, the quality of life inside and outside the workplace and a positive design perspective on technology have introduced new, broader views of HCI that require new roles for cognition. Table 4 provides an overview of the evolution of cognitive research in HCI.

Table 4. The Evolution of Cognitive Research in HCI

Time period	Context	Theory and Concepts	Areas of Interest	Examples
1970s to early 1980s	Transactions, data entry, reporting Office work, decision making	Psychological Type	Error messages Report design Information overload Individual differences	Cognitive style and information processing (Robey & Taggart, 1981) Graphs and color (Benbasat & Dexter, 1985)
1980s to early 1990s	Teamwork Searching information	Complexity theory TRA GOMS Cognitive fit	Static visual techniques	3D, windows and complexity (Te'eni, 1989) Abstraction Levels on Query Performance (Hock Chuan, Kwok Lee, & Keng Leng, 1993) Protocols of process (Todd & Benbasat, 1987)
1990s to early 2000s	The Web and Online Commerce Mobile	Cost-benefit TRA Social identity and de-individuation	Dynamic visual techniques Communication, virtual teams Usability Constructs related to emotion	Direct manipulation, 3D, Visualization and Animation (Zhang, 2000) Microcomputer playfulness and flow (Webster & Martocchio, 1992)
2000's	Social Media/Web 2.0 Tablet	Complexity theory Emotion and cognition TRA Social identity and de-individuation	Location sensitive Personalization Virtual reality (Cognitive) trust	Mobile ebooks Virtual worlds Personalization and Trust (Komiak & Benbasat, 2006) Emotion and cognition in website design – (Deng & Poole, 2010) Images in websites and cultural differences (Cyr et al., 2009)

5. Synthesis and Future Directions

5.1. From Enduring Questions to Cognitive Qualities

We explored three distinct streams of IS cognitive research and identified enduring questions that characterize the programs of research in these streams. We expect that they will persist, as enduring questions, though they may evolve and be re-interpreted. However, appreciating past contributions more broadly and illuminating new vistas for research require a holistic synthesis across these streams.

Building from the enduring questions, we sought to identify "Cognitive Qualities of IT" by exploring the common themes across the three streams and their enduring questions. We use the term cognitive qualities to refer to IT's ability to affect cognitive processes and ultimately performance. Cognitive qualities are, thus, design-relevant aspects of IT. This potential is not simply a matter of design and implementation, but design and implementation with an understanding of cognitive processes. Cognitive qualities explain the relationship between the design and implementation of the IT artifact and task performance. We identify four key cognitive qualities of IT: interactivity, fit, cooperativity, and affordances. We discuss these four cognitive qualities further below.

In both the HCI and DSS streams, we saw interactivity as a key cognitive quality. For example, DSS-RQ1's concern with the balance between computer-based models (representations) and user cognition came to the fore with the introduction of interactive DSS. In HCI, by definition, interactivity is a central idea. Intuitively, the notion has meaning. More formally, it is a subject of some debate. For example, Rafaeli and Sudweeks (1997, p. 1) define interactivity in the context of computer-mediated communication as "the extent to which messages in a sequence relate to each other, and especially the extent to which later messages recount the relatedness of earlier messages".

Featherman, Wright, Thatcher, Zimmer, and Pak (2011, p. 3) define interactivity "as a consumer's ability to manipulate objects in an online environment." Cognitively, interactivity implies a feedback loop of some form (e.g., in a sequence of messages, or in the manipulation of objects). Within development, the feedback loop of develop-test-refine has been present from the early days of programming. The desire for greater "interactivity" in development processes and tools is, thus, fundamental. For example, we can identify an interest in interactivity in the early comparisons of debugging in an online setting rather than off-line coding and batch submission (e.g., Sackman et al., 1968) -- which falls under the scope of DEV-RQ1. Thus, we find interactivity to be a key cognitive quality of IT evident in all three streams.

Across all streams, the cognitive quality of fit played an important role. For example, in the software development arena this was evident in work examining factors (tools, techniques, and practices) that reduce the cognitive complexity of development (under the scope of DEV-RQ1). Cognitive fit between a maintenance task and the application domain knowledge of the maintainer was found to enhance performance (Khatri et al., 2006; Shaft & Vessey, 2006), which falls under the scope of DEV-RQ2. Theoretically, poor cognitive fit leads to greater effort, due to the need to transform representations. Within the scope of HCI-RQ1 and HCI-RQ5, cognitive fit was initially developed to address the graphs versus table debate and has subsequently been applied to address other interface design issues (Adipat, Zhang, & Zhou, 2011). As an antecedent of effort, fit has bearing on the effort/accuracy work examined in conjunction with DSS-RQ2. Within the scope of DSS-RQ3, fit was explicitly introduced into the GDSS context (e.g., Dennis et al., 2001). Although the use of fit here was broader than just cognitive, a major factor was the cognitive nature of the tasks (e.g., judgment versus sensemaking) (see also Zigurs & Buckland, 1998). Whether the original concept of cognitive fit (Vessey & Galletta, 1991), or other cognitive perspectives on fit (e.g., see Davern, 2007, for a cognitive interpretation of Goodhue's (1995) Task/Technology Fit), it is an important cognitive quality of IT.

A common trend we observe is the shift from a focus on individual cognition to distributed cognition. We use the term "cooperativity" to describe the cognitive quality of IT that captures the extent to which IT supports and enables distributed cognition. Our introduction of the term is intended to emphasize the importance of collaboration (between an individual and an IS or among individuals, groups, and communities through an IS) while recognizing, like interactivity, it is a cognitively relevant characteristic of IT design.

Recall we defined distributed cognition (see Table 1) broadly to include knowledge sharing, collaborative task performance, shared mental models, shared cognition, and communication. Historically we see the growing focus on knowledge sharing and concern for supporting collaboration and groups (e.g., DEV-RQ3, DSS-RQ3, HCI-RQ3, and HCI-RQ5) as exemplifying the increased interest in design IT artifacts that exhibit cooperativity. The HCI literature is perhaps most advanced in this regard; explicitly recognizing distributed cognition as a foundation for research (Hollan et al., 2000), and with a large body of work in computer-mediated communication. However, there is scope for further cognitive work on distributed cognition in development (e.g., pair programming, distributed development teams), and in decision support (e.g., knowledge management, business intelligence). Given the centrality of collaboration here, a future research focus on cooperativity as a cognitive quality of IT is one that also needs to understand the broader social context of behavior with IS as well as its connection with other psychological factors such as emotion and attitudes (per HCI-RQ4). Across all streams, we observe IT enabling performance by introducing structure to the tasks and environment in which cognition takes place, whether that cognition is about development, decision making, or interface manipulation. This structuring is not simply a matter of IT design; it is design with an understanding of cognition – by our definition a cognitive quality of IT, which we label "affordances", borrowing from Ecological Psychology. Affordances reflect a complementarity between an individual and the environment (Gibson, 1979). A situation can afford a particular action for an individual with appropriate knowledge and abilities, and an individual can have the knowledge and abilities to carry out a particular action in an environment that affords such actions (Greeno, Moore, & Smith, 1993). As Kochevar (1994, p. 6) puts it: "Environments provide information structured to support specific behaviours, and adapted individuals are sensitive to such information patterns." We see evidence of this cognitive quality across the three research streams: for example, in the role of development techniques as providing affordances to facilitate building and maintenance of representations (per DEV-RQ1). Similarly, concepts like decisional guidance and system restrictiveness (which are encompassed by DSS-RQ2) are in effect about the design of the affordances for decision making with a DSS. Finally, HCI's concern for customizing and personalizing interfaces (HCI-RQ2) and supporting diverse users (HCI-RQ5) are quite pointedly matters of the design of affordances – the complementarity between interface and user. Indeed, the design imperative underlying all of the RQs can be expressed as an issue of the design of affordances. It is perhaps not surprising then that the term affordance has already appeared in the IS literature (e.g., Davern, 1996; Markus & Silver, 2008) and is quite well-known in the HCI literature (e.g., Norman, 1988).

In our synthesis across the three streams, we identified four cognitive qualities of IT: interactivity, fit, cooperativity, and affordances. We cannot claim that this list is exhaustive, as it is derived from our historical analysis and, therefore, limited by the scope of our history. Moreover, these cognitive qualities are not necessarily distinct, but are interrelated. For example, affordances may impact fit and interactivity, which, in turn, may influence cooperativity.

As the enduring questions provide a means to organize the research within each stream, the cognitive qualities provide a basis to examine the history (and future) of cognitive research in IS. Table 5 details the four cognitive qualities we identified and how they relate to IS contexts, and our conceptualization of IS as representations.

Table 5. The Cognitive Qualities of IT

Cognitive Quality	Examples of IT attributes	Representations	IS Contexts and Concepts
Interactivity	Fast response time, immediate feedback	Engaging with representations	Interactive decision support, social media and communications
Fit	Match between quantitative graphics and type of decision task	The relationship between representations	Cognitive Fit, task-technology fit
Cooperativity	Applications and connectivity to exchange graphics and photos, ability to translate messages	The sharing of representations	Group decision support, knowledge management systems, pair programming distributed development, transactive memory, shared mental models
Affordances	Query/ transaction by example, menus, scenarios	Representations as constraints and enablers	System restrictiveness, decisional guidance, systems development methodologies, information presentation format

As technology evolved, the context of use changed, and the manner in which cognitive qualities manifest themselves changed also. Table 6 describes this evolution for interactivity across the areas of research we examined. For example, DSS research examined the effect of interactivity on feedback designed to achieve more effective decision making and moved to considering enabling interactivity between multiple users. Similarly, in development, the IT-enabled interactivity evolved from batch, to online, to interactively enabling distributed development (such as geographically distributed designers working on a shared virtual representation of a system to be developed).

Table 6. Illustrating the Evolution of a Cognitive Quality of IT – Interactivity

Period	IT	Nature of Interactivity		
		SD	DSS	HCI
1960s	Mainframes	Programmers submit code and receive error report/execution results.	Decision makers get standardized reports	Only "operators" interact directly
1970s	Timesharing systems prevalent. The beginnings of desktop computing	Programmers code on-line and receive error feedback/ execution results in real-time.	Decision makers interact directly with DSS models (e.g., what-if scenarios).	User base expands, a broader base of users interact with systems. The introduction of more graphically based interactions.
1980s	Local Networking	Structured Development: Interacting with more modular code.	Decision makers begin to interact through DSS (GDSS)	Interacting with systems to find information.
1990s	The Web and Global Networking	Interacting with code produced by others. (Maintenance and Reuse) GUI development environments.	Interacting with knowledge-based systems through to knowledge management systems (interacting with the knowledge of others through IT)	Consumer and general public interactions with IT
2000s	Fully mobile computing/ social media	Distributed development (including open source): Interacting with others about code and design.	Interacting with the information environment (e.g., enterprise-wide BI systems) Interacting globally to share knowledge	Social interaction to develop collaborations, relationships, and communities of practice

5.2. The Future of Cognitive Research

As technology advances and new contexts of use emerge, we expect that the manner in which the cognitive qualities manifest themselves will evolve. Within the context of the three streams of research we analyzed, we see an exciting future in which the interplay among the different cognitive qualities provides an understanding of the development and use of IS in rich and meaningful contexts. For example, consider interactivity in conjunction with affordances and cooperativity: understanding and generating new knowledge from the interaction between people, between people and their environment, including the technologies with which they interact and how they structure the interactions between people, and people and the environment. Cognition will, thus, be distributed among people, organizations, and IT – an integration of knowledge residing in distributed agents that comes together in a hermeneutic dialog. The success of these combinations of interactivity, affordances, and cooperativity will very much be a matter of fit; the fit between system representations and mental representations in individuals or distributed across groups, and the fit between the representations of the task environment and these individual or distributed representations.

From an HCI perspective, we see these issues arising from the ubiquity of online information and the rapid communication enabled by social media. In the DSS context, we identified knowledge management systems and BI as areas in which cognitive studies are underrepresented. For development, understanding the distributed cognition in the context of open source development practices, and other contemporary development strategies, seems fertile ground.

Our view of the future of cognitive research is in contrast and complementary to the recent interest in NeuroIS: the application of neuroscience theories, methods, and tools in IS research (Dimoka, Pavlou, & Davis, 2007). Notably, the proposed research agendas for NeuroIS include all three subareas addressed in this manuscript (Dimoka et al., forthcoming; Dimoka et al., 2007; Riedl, Banker, Benbasat, Davis, & Dennis, 2010). NeuroIS and Cognitive IS research are complementary in much the same way as an understanding function of hardware complements and an understanding of software capabilities and performance (McClamrock, 1995). The two perspectives are at different levels of analysis. In historical terms, the rise of NeuroIS is somewhat intriguing, given early work in IS rejecting the value of neuroscience in informing design with Rao, Jacob, and Frank (1992, p. 149) providing "evidence against the use of the microscopic approach of neuroscience for providing theoretical foundations for DSS and human-machine interface design guidelines". Further they suggest, "the focus should instead be on psychological issues where cognitive functions are studied independently of their physical implementation. Research on cognitive functions should be carried out in terms of their high-level characteristics rather than micro-organization".

While Rao's critique may be overly harsh given the advances in neuroscience since 1992, it highlights the need to carefully distinguish the potential contributions of NeuroIS and cognitive research in IS. Whereas NeuroIS by technological and economic necessity operates at the individual level (imaging brain activity is expensive and a one-subject-at-a-time activity), we see distributed cognition as increasingly the foundation for cognitive IS research that goes beyond the individual and investigates richer and more contextualized environments.

The future direction of cognitive research in IS requires a shift in the mix of methods employed from the almost universal experimental study of cognition to the incorporation of field work. On the one hand, this can imply the sort of cognitive anthropology best illustrated by Hutchins (1995), but not exclusively so (see Davern, Mantena, & Stohr, 2008, for an example of the use of cognitive theory to explore archival field data). It does not spell the demise of the experimental approach, only that we more effectively capture the richness of the environment in the lab (while recognizing the concomitant challenges to internal validity). As in the past, IS research can draw from cognitive psychology for insight. For example, the field of ecological psychology, which still makes abundant use of experiments, views behavior formally as an emergent product of the interaction between an individual and his or her environment (Gibson, 1979). The challenge is developing further IS cognitive theories that can adequately capture the richness of the information environment we now operate in, and also reflect the dynamics of emergent behaviors. Inevitably, this will require a broader view of cognition and recognizing its connections to other drivers of behavior such as emotion and attitudes.

In addition to investigating the role of more diverse contexts, another fruitful area for IS researchers could be developing technologies and techniques based on cognitive theories. Cognitive researchers in IS have frequently examined new technologies and methods. They have less frequently used cognitive theory to drive the creation of tools and techniques. There are examples of such work. Kim et al. (2000) created a new diagramming technique based upon cognitive theory. Similarly, using the GDM, Buton-Jones and Meso (2006) created and compared different modeling grammars. Such efforts could span the DSS and HCI arenas. As "design research" gains momentum, cognitively-driven design research could be an important direction for cognitive IS research. Certainly, the enduring questions we consider across the three streams reflect an underlying intent to provide guidance to design. Moreover, the cognitive qualities of IT we have identified provide potentially powerful lenses for cognitive theory-driven design research. Indeed, we see the design task as operationalizing and implementing the different cognitive qualities of IT we have identified.

5.3. Limitations and Future Histories

In this historical analysis, we concentrated on the development of the substantive content of cognitive research in IS. We showed how technological advancements in the cognitive qualities of IT and the changing and expanding context of how the technology is used directed the research issues studied. We did not attempt to examine the motivations, practices, or social contexts of the researchers. One could argue that an analysis of networks of researchers may also contribute to understanding why and how cognitive research evolved. For instance, many IS researchers participated in conferences and published in journals outside IS (in our analysis we focused on mainstream IS journals, except for the earliest work when such journals did not exist). However, researchers of cognition in IS may participate in a variety of conferences as well as publish in specialized and non-IS journals. Specialized conferences and publications differ from IS conferences and publications in the requirements and expectations of the role of theory, method, and contribution in research. It might be interesting to examine the impact of this duality on researchers and their published work. A co-citation analysis, for instance, might help us understand if research in the specialized outlets has influenced the research of others in mainstream IS, and vice versa, and whether the researchers common to both networks serve as boundary spanners to spread the word in both directions. We encourage future accounts to consider this complementary perspective.

6. Conclusion

Our history of cognitive research in IS demonstrates that cognition topics have been a significant focus in IS research. Reflecting on this history, we exposed some of the influences on the evolution of cognitive research in IS across three areas: software development, decision support, and human-computer interaction. Just as it is clear that cognitive research has made a substantial contribution to IS, it is also clear that it has potential for future contribution. This is evident in the enduring questions we identified and in the cognitive qualities of IT we described above. Furthermore, extrapolating from our historical analysis, we postulate a robust future for cognitive research in IS: a future that is more contextually bound and, thus, even more relevant; a future that is more dynamic and, thus, more challenging. We call on the field, and cognitive researchers in IS, in particular, to embrace that challenge.

References

Adelson, B. (1981). Problem solving and the development of abstract categories in programming languages. Memory and Cognition, *9*, 422-433.

Adelson, B., & Solloway, E. (1985). The role of domain experience in software design. *IEEE Transactions on Software Engineering*, *11*(11), 1351-1359.

Adipat, B., Zhang, D., & Zhou, L. (2011). The effects of tree-view based presentation adaptation on model web browsing. *Management Information Systems Quarterly*, *35*(1), 99-121.

Agarwal, R., Sinha, A., & Tanniru, M. (1996). Cognitive fit in requirements modelling: A study of object and process methodologies. *Journal of Management Information Systems*, *13*(2), 137-162.

Ajzen, I., & Fishbein, M. (1980). *Understanding attitudes and predicting social behavior*. Englewood Cliffs, NJ: Prentice-Hall.

Alavi, M., & Leidner, D. E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *Management Information Systems Quarterly*, *25*(1), 107-136.

Alter, S. (1977). A taxonomy of decision support systems. *Sloan Management Review*, *19*(1), 37-56.

Anderson, J. R. (1995). *Cognitive Psychology and its Implications*. New York: W. H. Freeman and Company.

Arnold, V., Clark, N., Collier, P. A., Leech, S. A., & Sutton, S. G. (2006).The differential use and effect of knowledge-based system explanations in novice and expert judgment decisions. *Management Information Systems Quarterly*, *30*(1), 79-97.

Arnold, V., & Sutton, S. (1998). The theory of technology dominance: Understanding the impact of intelligent decision aids on decision makers' judgments. *Advances in Accounting Behavioural Research, 1*, 175-194.

Arnott, D., & Pervan, G. (2005). A critical analysis of decision support systems research. *Journal of Information Technology*, *20*, 67-87.

Asch, S. E. (1952). *Social Psychology*. Englewood Cliffs, N.J.: Prentice-Hall.

Avital, M., & Te'eni, D. (2009). From generative fit to generative capacity: Exploring an emerging dimension of information systems design and task performance. *Information Systems Journal*, *19*(4), 345-367.

Babu, R., Singh, R., & Ganesh, J. (2010). Understanding blind users' web accessibility and usability problems. *AIS Transactions on Human-Computer Interaction*, *2*(3), 73-94.

Balijepally, V., Mahapatra, R., Nerur, S., & Price, K. H. (2009). Are two heads better than one for software development? The productivity paradox of pair programming. *Management Information Systems Quarterly*, *33*(1), 91-118.

Bariff, M. L., & Lusk, E. J. (1977).Cognitive and personality tests for the design of management information systems. *Management Science*, *23*(8), 820-829.

Barkhi, R. (2002). Cognitive style may mitigate the impact of communication mode. *Information and Management*, *39*(8), 677-688.

Benbasat, I., & Dexter, A. S. (1985). An experimental evaluation of graphical and color-enhanced information presentation. *Management Science*, *31*(11), 1348-1364.

Benbasat , I. and Schroeder, R. (1977). An experimental investigation of some MIS design variables. Management Information Systems Quarterly, 1(1), 37-49

Benbasat, I., & Taylor, R. N. (1978). The impact of cognitive styles on information system design. *Management Information Systems Quarterly*, *2*(2), 43-54.

Benbasat, I., & Todd, P. (1996).The effects of decision support and task contingencies on model formulation: A cognitive perspective. *Decision Support Systems*, *17*(4), 241-252.

Benbasat, I., & Zmud, R. W. (2003). The identity crisis within the IS discipline: Defining and communicating the discipline's core properties. *Management Information Systems Quarterly*, *27*(2), 183-194.

Bergman, M., Lyytinen, K., & Mark, G. (2007). Boundary objects in design: An ecological view of design artifacts. *Journal of the Association for Information Systems*, *8*(11), 546-568.

Berlyne, D. E. (1971). *Aesthetics and psychobiology*. New York, NY: Appleton-Century-Crofts.

Blattberg, R. C., & Hoch, S. J. (1990). Database models and managerial intuition: 50% model + 50% manager. *Management Science*, *36*(8), 887-899.

Bodart, F., Patel, A., Sim, M., & Weber, R. (2001). Should optional modeling be used in conceptual modeling? A theory and three empirical tests. *Information Systems Research*, *12*(4), 384-405.
Boland, R. J. J., Tenkasi, R. V., & Te'eni, D. (1994). Designing information technology to support distributed cognition. *Organization Science*, *5*(3), 456-475.
Bonczek, R. H., Holsapple, C. W., & Whinston, A. B. (1981). *Foundations of Decision Support Systems.* New York: Academic Press.
Brittain White, K. (1984). A preliminary investigation of information systems team structures. *Information and Management*, *7*(6), 331-335
Brittain White, K., & R. Leifer. (1986). Information systems development success: Perspectives from project team participants. *Management Information Systems Quarterly*, *10*(3), 215-223.
Brooks, F. P. (1987). No silver bullet: Essence and accidents of software engineering. *Computer*, 20(4), 10-19.
Brooks, R. E. (1977). Towards a theory of the cognitive processes in computer programming. *International Journal of Man-Machine Studies*, *9*(6), 737-751.
Browne, G., & Rogich, M. (2001). An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques. *Journal of Management Information Systems*, *17*(4), 223-249.
Bui, T., & Loebbecke, C. (1996). Supporting cognitive feedback using system dynamics: A demand model of the global system of mobile telecommunication. *Decision Support Systems*, *17*(2), 83-98.
Burton-Jones, A., & Meso, P. (2006). Conceptualizing systems for understanding: An empirical test of decomposition principles in OO analysis. *Information Systems Research*, *17*(1), 38-60.
Burton-Jones, A., & Meso, P. (2008). The effects of decomposition quality and multiple forms of information on novices' understanding of a domain from a conceptual model. *Journal of the Association for Information Systems*, *9*(12), 748-802.
Card, S., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Carroll, J. M. (2010). Beyond being social: Prospects for transformative social computing. *Communications of the Association for Information Systems*, *27*(34), 641-650
Chakraborty, S., Sarkar, S., & Sarker, S. (2010). An exploration into the process of requirements elicitation: A grounded approach. *Journal of the Association for Information Systems*, *4*(2), 212-219.
Chervany, N. L., & Dickson, G. W. (1974). An experimental evaluation of information overload in a production environment. *Management Science*, *20*(10), 1335-1344
Chilton, M., Hardgrave, B., & Armstrong, D. (2005). Person-job cognitive style fit for software developers: The effect of strain and performance. *Journal of Management Information Systems*, *22*(2), 193-226.
Chu, P. C., & Spires, E. E. (2000). The joint effects of effort and quality on decision strategy choice with computerized decision aids. *Decision Sciences*, *31*(2), 259-292.
Cooksey, R. W. (1996). *Judgment analysis: Theory, methods, and applications*. San Diego: Academic Press, Inc.
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.
Cyr, D., Head, M., Larios, H., & Pan, B. (2009). Exploring human images in website design: A multi-method approach. *Management Information Systems Quarterly*, *33*(3), 530-566.
Daft, R. L., & Lengel, R.H. (1986). Organizational information requirements, media richness and structural design. *Management Science*, *32*(5), 554-571.
Davern, M. J. (1996). When Good Fit is Bad: The Dynamics of Perceived Fit. *Proceedings of the 17th International Conference on Information Systems, Cleveland, Ohio, 112-121.*
Davern, M. J. (2007). Towards a unified theory of fit: Task, technology and individual. In D. Hart & S. Gregor (Eds.), *Information systems foundations: Theory, representation and reality* (pp. 49-69). Acton, Australia: ANU E-Press.
Davern, M., & Kamis, A. (2010). Knowledge matters: Restrictiveness and performance with decision support. *Decision Support Systems*, *49*(4), 343-353.

Davern, M. J. & Parkes, A. (2010). Incommensurability in design science: Which comes first theory or artifact? In S. Gregor & D. Hart (Eds.), *Information Systems Foundations: The Role of Design Science* (pp. 75-90). Acton, Australia: ANU E Press.
Davern, M. J., Mantena, R., & Stohr, E. A. (2008). Diagnosing decision quality. *Decision Support Systems*, *45*, 123-139.
Davenport, T., & Prusak, L. (1998). *Working knowledge: How organizations manage what they know*. Boston, MA: Harvard Business School Press.
Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *Management Information Systems Quarterly, 13*(3), 319-340.
Davis, F. D., & Kottemann, J. E. (1995). Determinants of decision rule use in a production planning task. *Organizational Behavior and Human Decision Processes*, *63*(2), 145-157.
Dawande, M., Johar, M., Kumar, S., & Mookerjee, V. S. (2008). A comparison of pair versus solo programming under different objectives: An analytical approach. *Information Systems Research*, *19*(1), 71–92.
Deng, L., & Poole, M. S. (2010). Affect in web interfaces: A study of the impacts of web page visual complexity and order. *Management Information Systems Quarterly*, *34*(4), 711-730.
Dennis, A. R., Wixom, B. H., & Vandenberg, R. J. (2001). Understanding fit and appropriation effects in group support systems via meta-analysis. *Management Information Systems Quarterly*, *25*(2), 167-194.
De Sanctis, G., & Gallupe, R. B. (1987).A foundation for the study of group decision support systems. *Management Science*, *33*(5), 589-609.
Détienne, F. (2002). *Software design – cognitive aspects*. London: Springer.
Dhaliwal, J. S., and Benbasat, I. (1996). The use and effects of knowledge-based system explanations: Theoretical foundations and a framework for empirical evaluation. *Information Systems Research*, *7*(3), 342-362.
Dickmeyer, N. (1983). Measuring the effects of a university planning decision aid. *Management Science*, *29*(6), 673-685.
Dickson, G. W., Senn, J. A., & Chervany, N. L. (1977). Research in management information systems: The Minnesota experiments. *Management Science*, *23*(9), 913-923.
Dimoka, A., Banker, R. D., Benbasat, I., Davis, F. D., Dennis, A. R., Gefen, D., Gupta, A., Ischebeck, A., Kenning, P., Pavlou, P. A., Müller-Putz, G., Riedl, R., vom Brocke, J., & Weber, B. (forthcoming). On the use of neurophysiological tools in IS research: Developing a research agenda for neuroIS. *MIS Quarterly*.
Dimoka, A., Pavlou, P., & Davis, F. (2007). *NeuroIS: The potential of cognitive neuroscience for information systems research.* Paper presented at the International Conference on Information Systems (ICIS), Montreal, Canada.
Duchessi P., & O'Keefe, R. M. (1992). Contrasting successful and unsuccessful expert systems. *European Journal of Operational Research*, *61*, 122-134
Duchessi, P., & O'Keefe, R. M. (1995). Understanding expert systems success and failure. *Expert Systems with Applications*, *9*(2), 123-133.
Eining, M., & Dorr, P. B. (1991). The Impact of expert system usage on experiential learning in an auditing setting. *Journal of Information Systems*, *5*(1), 1-16.
Elbashir, M. Z., Collier, P. A., & Davern, M. J. (2008).Measuring the effects of business intelligence systems: The relationship between business process and organizational performance. *International Journal of Accounting Information Systems*, *9*(3), 135-153.
Espinosa, J. A., Slaughter, S. A., Kraut, R. E., & Herbsleb, J. D.(2007). Team knowledge and coordination in geographically distributed software development. *Journal of Management Information Systems*, *24*(1),135-169.
Farwell, D. C. (1984). DSS Mechanisms for judgemental flexibility: An exploratory survey. *Journal of Management Information Systems*, *1*(2), 72-79.
Featherman, M., Wright, R. T., Thatcher, J., Zimmer, J. C., & Pak, R. (2011). The influence of interactivity on e-service offerings: An empirical examination of benefits and risks. *AIS Transactions on Human-Computer-Interaction*, *3*(1), 1-25.
Gallupe, R. B., Cooper, W. H., Grise, M. L., & Bastianutti, L. M. (1994). Blocking electronic brainstorms. *Journal of Applied Psychology*, *79*(1), 77-86.

Gane, T., & Sarson, C. (1979). *Structured system analysis: Tools and techniques*. Englewood Cliffs, NJ: Prentice-Hall.
Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*(2), 155-170.
Gentner, D., & Stevens, A. L. (Eds.) (1983). *Mental models*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Gibson, J. J. (1979). *An ecological approach to visual perception*. Boston, MA, Houghton Mifflin.
Goodhue, D. L. (1995). Understanding user evaluations of information systems. *Management Science*, *41*(12), 1827-1844.
Goodhue, D.L., & Thompson, R. L. (1995).Task-technology fit and individual performance. *MIS Quarterly*,*19*(2), 213-236.
Goodwin, P. (2005). Providing support for decisions based on time series information under conditions of asymmetric loss. *European Journal of Operational Research*, *163*(2), 388-402.
Goodwin, P., Fildes, R., Lawrence, M., & Stephens, G. (2011). Restrictiveness and guidance in support systems. *Omega*, *39*(3), 242-253.
Gorla, N., & Lam, Y. W. (2004). Who should work with whom? Building effective software project teams. *Communication of the ACM*, *47*(6), 79-82.
Goswami, S., Chan, H. C., & Kim, H.-W. (2008). The role of visualization tools in spreadsheet error correction from a cognitive fit perspective. *Journal of the Association for Information Systems*, *9*(6), 321-343.
Greeno, J. G., Moore, J. L., & Smith, D. R. (1993) Transfer of situated learning. In D. K. Detterman & R. J. Sternberg (Eds.), *Transfer on trial: Intelligence, cognition and instruction* (pp. 99-167). Norwood, NJ: Ablex Publishing Co.
Gregor, S. (2001). Explanations from knowledge-based systems and cooperative problem solving: An empirical study. *International Journal of Human-Computer Studies*, *54*(1), 81-105.
Gregor, S., & Benbasat, I. (1999). Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly*, *23*(4), 497-530.
Grise, M., & Gallupe, R. B. (2000). Information overload: Addressing the productivity paradox in face-to-face electronic meetings. *Journal of Management Information Systems*, *16*(3), 157-185.
Gureckis, T. M., & Goldstone, R. L. (2006). Thinking in groups. *Pragmatics and Cognition*, *14*, 293-311.
Hampton, C. (2005). Determinants of reliance: An empirical test of the theory of technology dominance. *International Journal of Accounting Information Systems*, *6*(4), 217-240.
Harzing, A. W. (2011). *Publish or perish* (Ver. 3.139). Available from www.harzing.com/pop.htm
Henderson, J. C., & Nutt, P. C. (1980).The influence of decision style on decision making behavior. *Management Science*, *26*(4), 371-386.
Hendriks, P. H. J., & Vrien, D. J. (1999). Knowledge-based systems and knowledge management: Friends or foes? *Information and Management*, *35*(2), 113-125.
Hess, T. J., Fuller, M., & Campbell, D. E. (2009). Designing interfaces with social presence: Using vividness and extraversion to create social recommendation agents. *Journal of the Association for Information Systems*, *10*(12), 889-919.
Hoch, S. J., & Schkade, D. A. (1998). A psychological approach to decision support systems. *Management Science*, *42*(1), 51-64.
Hock Chuan, C., Kwok Kee, W., & Keng Leng, S. (1993). User-database interface: The effect of abstraction levels on query performance. *MIS Quarterly*, *17*(4), 441-464.
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.
Huber, G. P. (1983). Cognitive style as a basis for MIS and DSS designs: Much ado about nothing? *Management Science*, *29*(5), 567-579.
Hutchins, E. (1990). The technology of team navigation. In J. R. Galegher, R. E. Kraut, & C. Egido (Eds.), *Intellectual teamwork: Social and technological foundations of cooperative work* (pp. 191-220). Hillsdale, NJ: L. Erlbaum Associates
Hutchins, E. (1991). Organizing work by adaptation. *Organization Science*, *2*(1), 14-39.
Hutchins, E. (1995). *Cognition in the Wild*. Cambridge, Mass.: The MIT Press.

Huysmans, J. H. B. M. (1970). The effectiveness of the cognitive-style constraint in implementing operations research proposals. *Management Science*, *17*(1), 92-104.
Irwin, G. (2002). The role of similarity in the reuse of object oriented analysis models. *Journal of Management Information Systems*, *19*(2), 221-250.
James, W. (1950). *The principles of psychology*. New York: Dover Publications. (Original work published 1890)
Jeffries, R., Turner, A. A., Polson, P. G., & Atwood, M. E. (1981). The processes involved in designing software. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 255-284). Hillsdale, NH: L. Erlbaum Associates.
Jiang, J.J., & Klein, G. (2000). Side effects of decision guidance in decision support systems. *Interacting with Computers*,*12*(5), 469-481.
Johnson, E. J., & Payne, J. W. (1985). Effort and accuracy in choice. *Management Science*, *31*(4), 395-414.
Jones, D. R., & Brown, D. (2002). The division of labor between human and computer in the presence of decision support system advice. *Decision Support Systems*, *33*, 375-388
Kaiser, K., & Bostrom, R. (1982). Personality characteristics of MIS project teams: An empirical study and action-research design. *MIS Quarterly*, *6*(4), 43-60.
Kamis, A., Koufrais, M., & Stern, T. (2008) Using an attribute-based decision support system for user-customized products online: An experimental investigation. *MIS Quarterly*, *32*(1), 159-177.
Kayande, U., De Bruyn, A., Lilien, G. L., Rangaswamy, A., & van Bruggen, G. H. (2009). How incorporating feedback mechanisms in a DSS affects DSS evaluations*. Information Systems Research*, *20*(4), 527-546.
Keen, P. G. W. (1980). MIS research: Reference discipline and a cumulative tradition. *Proceedings of the first international conference on information systems, Philadelphia, PA, 9-18,*
Keen, P., & Scott-Morton, M. (1978). *Decision support systems: An organizational perspective*. Reading, MA: Addison – Wesley Inc.
Kennedy, M., Te'eni, D., & Treleavan, J. (1998). Impacts of decision task, data and display on strategies for extracting information. *International Journal of Human Computer Studies*, *48*(2), 159-180.
Khatri, V., Vessey, I., Ramesh, V., Clay, P., & Park, S.-J. (2006). Understanding conceptual schemas: Exploring the role of application domain and IS domain knowledge. *Information Systems Research*, *17*(1), 81-99.
Kim, J., & Lerch, F. J. (1997). Why is programming (sometimes) so difficult? Programming as scientific discovery in multiple problem spaces. *Information Systems Research*, *8*(1), 25-50.
Kim, J., Hahn, J., & Hahn, H. (2000). How do we understand a system with (so) many diagrams? Cognitive integration processes in diagrammatic reasoning. *Information Systems Research*, *11*(3), 284-303.
Kleinmuntz, B. (1990). Why we still use our heads instead of formulas: Toward an integrative approach. *Psychological Bulletin*, *107*, 296-310.
Kochevar, L. K. (1994). *Generativity of expertise* (Unpublished doctoral dissertation). University of Minnesota, Minnesota.
Kohavi, R., Rothleder, N. J., & Simoudis, E. (2002). Emerging trends in business analytics. *Communications of the ACM*, *45*(8), 45-48.
Komiak, S. Y. X., & Benbasat, I. (2006).The effects of personalization and familiarity on trust and adoption of recommendation agents. *Management Information Systems Quarterly*, *30*(4), 941-960.
Kottemann, J. E., Davis, F. D., & Remus, W. E. (1994). Computer-assisted decision making: Performance, beliefs, and the illusion of control. *Organizational Behavior and Human Decision Processes*, *57*(1), 26-37.
Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth 10,000 words. *Cognitive Science*, *11*(1), 65-99.
Letovsky, S. (1986). Cognitive processes in program comprehension. In E. Soloway & S. Iyengar (Eds.), *Empirical studies of programmers* (pp. 58-79). Norwood, NJ: Ablex Publishing.
Lusk, E. J., & Kersnick, M. (1979). The effect of cognitive style and report format on task performance: The MIS design consequences. *Management Science*, *25*(8), 787-798.

Mackay, J. M., & Elam, J. J. (1992). A comparative study of how experts and novices use a decision aid to solve problems in complex knowledge domains. *Information Systems Research, 3*(2), 150-172.
Majchrzak, A., Beath, M., & Lim, R. (2005). Managing client dialogues during information systems design to facilitate client learning. *MIS Quarterly*, *29*(4), 653-672.
Mao, J-Y., & Benbasat, I. (2000). The use of explanations in knowledge-based systems: Cognitive perspective and a process-tracing analysis. *Journal of Management Information Systems*, *17*(2), 153-179.
Marakas, G., & Elam, J. (1998). Semantic structuring in analyst acquisition and representation of facts in requirements analysis. *Information Systems Research*, *9*(1), 37-63.
Markus, M. L., & Silver, M. S. (2008). A foundation for the study of it effects: A new look at Desanctis and Poole's concepts of structural features and spirit. *Journal of the Association for Information Systems*, *9*(10), 609-632.
Mascha, M. F., & Smedley, G. (2007). Can computerized decision aids do "damage"? A case for tailoring feedback and task complexity based on task experience. *International Journal of Accounting Information Systems*, *8*(2), 73-91.
Mason, R. O., & Mitroff, I. I. (1973). A program for research on management information systems. *Management Science*, *19*(5), 475-487.
Mayer, R. E. (2001). *Multimedia learning*. New York, NY: Cambridge University Press.
McClamrock, R. (1995). *Existential cognition: computational minds in the world*. Chicago: University of Chicago Press.
McKeithen, K. B., Reitman, J. S., Rueter, H. H., & Hirtle, S. C. (1981). Knowledge organization and skill differences in computer programmers. *Cognitive Psychology*,*13*, 307-325.
McIntyre, S. H. (1982). An experimental study of the impact of judgment-based marketing models. *Management Science*, *28*(1), 17-33.
Meehl, P. (1954). *Clinical versus statistical prediction*. Minneapolis, MN: University of Minnesota Press.
Mennecke, B. E., Crossland, M. D., & Killingsworth, B. L. (2000). Is a map more than a picture? The role of SDSS technology, subject characteristics, and problem complexity on map reading and problem solving. *MIS Quarterly*, *24*(4), 601-629.
Metaxiotis, K., & Psarras, J. (2003). Neural networks in production scheduling: Intelligent solutions and future promises. *Applied Artificial Intelligence*, *17*(4), 361-373.
Miranda, S. M., & Saunders, C. S. (2003). The social construction of meaning: An alternative perspective on information sharing. *Information Systems Research*, *14*(1), 87-101.
Moran, T. P. (1981). The command language grammar: A representation for the user of interactive interface of computer systems. *International Journal of Man-Machine Studies*, *15*(1), 3-50.
Nah, F. F. H., & Benbasat, I. (2004). Knowledge-based support in a group decision making context: An expert-novice comparison. *Journal of the Association for Information Systems*, *5*(3), 125-150.
Neisser, U. (1967). *Cognitive Psychology*. New York, NY: Appleton-Century-Crofts.
Neisser, U. (1976). *Cognition and reality: Principles and implications of cognitive psychology*. San Francisco, CA: W.H. Freeman and Co.
Nelson, H. J., Armstrong, D. J., & Nelson, K. M. (2009). Patterns of transition: The shift from traditional to object-oriented development. *Journal of Management Information Systems*, *25*(4), 271-297.
Nelson, K. M., Nadkarni, S., Narayanan, V. K., & Ghods, M. (2000). Understanding software operations support expertise: A revealed causal mapping approach. *Management Information Systems Quarterly*, *24*(3), 475-507.
Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, N.J., Prentice-Hall.
Nonaka, I. (1991). The knowledge creating company. *Harvard Business Review*, *69*, 96-104.
Norman, D. A. (1988). *The psychology of everyday things*. New York: Basic Books.
Nunamaker Jr, J. F., Briggs, R. O., Mittleman, D. D., Vogel, D. R., & Balthazard, P. A. (1997). Lessons from a dozen years of group support systems research: A discussion of lab and field findings. *Journal of Management Information Systems*, *13*(3), 163-207.
Nutt, P.C. (1986). Evaluating MIS design principles. *Management Information Systems Quarterly*, *10*(2), 139-156.

Olson, G. M., & Olson, J. S. (2000). Distance matters. *Human-Computer Interaction*, *2/3*(15), 139-178.
Orlikowski, W. J., & Iacono, C. S. (2001). Research Commentary: Desperately seeking the "IT" in IT research - a call to theorizing the IT artifact. *Information Systems Research*, *12*(2), 121-134.
Paradice, D. B., & Courtney, J. F. (1986). Controlling bias in user assertions in expert decision support systems for problem formulation. *Journal of Management Information Systems*, *3*(1), 52-64.
Parikh, M., Fazlollahi, B., & Verma, S. (2001). The effectiveness of decisional guidance: An empirical evaluation. *Decision Sciences*, *32*(2), 303-331.
Payne, J. W. (1976). Task complexity and contingent processing in decision making: An information search and protocol analysis. *Organizational Behavior and Human Performance*, *16*(2), 366-387.
Payne, J. W., Bettman, J. R., & Johnson, E. J. (1993). *The adaptive decision maker*. Cambridge: Cambridge University Press
Pennington, N. (1987). Stimulus structures and mental representations in expert comprehension of computer programs. *Cognitive psychology*, *19*, 295-341.
Perrig, W., & Kintsch, W. (1985). Proposition and situational representations of text. *Journal of Memory and Language*, *24*, 503-518.
Phillips-Wren, G., Mora, M., Forgionne, G., & Gupta, J. (2009). An integrative evaluation framework for intelligent decision support systems. *European Journal of Operational Research*,*195*, 642-652.
Potter, R. E., & Balthazard, P. (2004). The role of individual memory and attention processes during electronic brainstorming. *MIS Quarterly*, *28*(4), 621-643.
Power, D. J. (2007). *A brief history of decision support systems*. Retrieved from http://dssresources.com/history/dsshistory.html
Pracht, W. E., & Courtney, J. F. (1988). The effects of an interactive graphics-based DSS to support problem structuring. *Decision Sciences*, *19*(3), 598-621.
Prietula, M. J., & March, S. A. (1991). Form and substance in physical database design: An empirical study. *Information Systems Research*, *2*(4), 287-314.
Ramaprasad, A. (1987). Cognitive process as a basis for MIS and DSS design. *Management Science*, *33*(2), 139-148.
Rafaeli, S., & Sudweeks, F. (1997). Networked interactivity. *Journal of Computer-Mediated Communication*, *2*(4).
Rao, H. R., Jacob, V. S., & Frank, L. (1992). Hemispheric specialization, cognitive differences, and their implications for the design of decision support systems. *MIS Quarterly*, *16*(2), 145-151.
Riedl, R., Banker, R., Benbasat, I., Davis, F., & Dennis, A. (2010). On the foundations of neuroIS: Reflections on the Gmunden Retreat 2009. *Communications of the Association for Information Systems, 27*, 243-264.
Rist, R. (1989). Schema creation in programming. *Cognitive Science*,*13*, 389-414.
Robey, D., & Taggart, W. (1981). Measuring manager's minds: The assessment of style in human information processing. *Academy of Management Review*, *6*(3), 375-383.
Robey, D., & Taggart, W. (1982). Human information processing in information and decision support systems. *MIS Quarterly*, *6*(2), 61-73.
Sackman, H., Erikson, W. J., & Grant, E. E. (1968). Exploratory experimental studies comparing on-line and off-line programming performance. *Communications of the ACM*, *1*, 3-11.
Sage, A. P. (1981). Behavioral and organizational considerations in the design of information systems and processes for decision and planning support. *IEEE Transactions on Systems, Man and Cybernetics*,*11*(9), 640-678.
Salzinger, K. (2005). Clinical, statistical, and broken-leg predictions. *Behavior and Philosophy*, *33*, 91-99.
Santhanam, R. L., Seligman, L., & Kang, D. (2007). Postimplementation knowledge transfers to users and Information Technology professionals. *Journal of Management Information Systems*, *24*(1), 171-199.
Schenk, K., Vitalari, N., & Davis, K. (1998). Differences between novice and expert systems analysts: What do we know and what do we do? *Journal of Management Information Systems*, *15*(1), 9-50.

Shaft, T. M., & Vessey, I. (1995). The relevance of application domain knowledge: The case of computer program comprehension. *Information Systems Research*, *6*(3), 286-299.

Shaft, T. M., & Vessey, I. (1998). The relevance of application domain knowledge: Characterizing the computer program comprehension process. *Journal of Management Information Systems*, *15*(1), 51-78.

Shaft, T. M., & Vessey, I. (2006). The role of cognitive fit in the relationship between software comprehension and modification. *MIS Quarterly*, *30*(1), 29-55.

Shanks, G., Tansley, E., Nuredini, J., Tobin, D., & Weber, R. (2008). Representing part-whole relations in conceptual modeling: An empirical evaluation. *Management Information Systems Quarterly*, *32*(3), 553-573.

Sheetz, S. D., Irwin, G., Tegarden, D. P., Nelson, H. J., & Monarchi, D. E. (1997). Exploring the difficulties of learning object-oriented technology. *Journal of Management Information Systems*, *14*(2), 103-131.

Sheil, B. A. (1981). The psychological study of programming. *Computing Surveys*, *13*(1), 101-120.

Sheppard, S. B., Curtis, B., Milliman, P., & Love, T. (1979). Modern coding practices and programmer performance. *Computer*, *12*, 41-49.

Shneiderman, B. (1976). Exploratory experiments in programmer behavior. *International Journal of Parallel Programming*, *5*(2), 123-143.

Shneiderman, B. (1979). Human factors experiments in designing interactive systems. *IEEE Computer*, *12*(12), 9-19.

Shneiderman, B., & Mayer, R. (1979). Syntactic/semantic interactions in programmer behavior: A model and experiment results. *International Journal of Computer and Information Systems*, *8*(3), 10-24.

Shneiderman, B., Mayer, R., McKay, D., & Heller, P. (1977). Experimental investigations of the utility of detailed flowcharts in programming. *Communications of the ACM*, *20*(6), 373-381.

Short, J., Williams, E., & Christie, B. (1976). *The social psychology of telecommunications*. New York, NY: Wiley.

Silver, M. S. (1988). User perceptions of decision support system restrictiveness: An experiment. *Journal of Management Information Systems*, *5*(1), 51-65.

Silver, M. S. (1990). Decision support systems: Directed and non-directed change. *Information Systems Research*, *1*(1), 47-70.

Simon, H. (1955). A behavioral model of rational choice. *Quarterly Journal of Economics*, *1*(69), 99-118.

Simon, H. A. (1978). On how to decide what to do. *Bell Journal of Economics*, *9*(2), 494-507.

Singh, D. T. (1998). Incorporating cognitive aids into decision support systems: The case of the strategy execution. *Decision Support Systems*, *24*(2), 145-163.

Solloway, E., & Ehrlich, K. (1984). Empirical Studies of Programming Knowledge*. IEEE Transactions on Software Engineering, 10*(5), 595-609.

Song, J., Jones, D., & Gudigantala, N. (2007). The effects of incorporating compensatory choice strategies in web-based consumer decision support systems. *Decision Support Systems*, *43*(2), 359-374.

Sprague, R. H. J. (1980). A framework for the development of decision support systems. *MIS Quarterly*, *4*(4), 1-26.

Sprague, Jr., R. H., & Carlson, E. D. (1982). *Building effective decision support systems*. Englewood Cliffs, NJ: Prentice-Hall.

Srinivasan, A., & Te'eni, D. (1995). Modeling as a constrained problem solving process. *Management Science*, *41*(3), 419-434.

Tan, M. (1994). Establishing mutual understanding in systems design: An empirical study. *Journal of Management Information Systems*,*10*(4), 159-182.

Tan, C. H., Teo, H. H., & Benbasat, I. (2010).Assessing screening and evaluation decision support systems: A resource-matching approach. *Information Systems Research*, *21*(2), 305-326.

Te'eni, D. (1989). Determinants and consequences of perceived complexity in human-computer interaction. *Decision Sciences*, *20*, 166-181.

Te'eni, D. (1991). Feedback in DSS as a source of control: Experiments with the timing of feedback. *Decision Sciences*, *22*(3), 644-655.

Te'eni, D. (2001). A cognitive-affective model of organizational communication for designing IT. *MIS Quarterly*, *25*(2), 251-312.

Te'eni D., Carey, J., & Zhang, P. (2007). H*uman-computer interaction: Developing effective organizational information systems*. Hoboken, NJ: John Wiley and Sons.

Thornton, C. (2009), Representation recovers information. *Cognitive Science*, *33*, 1383–1412.

Todd, P., & Benbasat, I. (1987). Process tracing methods in decision support systems research: Exploring the black box. *Management Information Systems Quarterly*, *11*(4), 493-512.

Todd, P., & Benbasat, I. (1991). An experimental investigation of the impact of computer based decision aids on decision making strategies. *Information Systems Research*, *2*(2), 87-115.

Todd, P., & Benbasat, I. (1992). The use of information in decision making: An experimental investigation of the impact of computer-based decision aids. *Management Information Systems Quarterly*, *16*(3), 373-393.

Todd, P., & Benbasat, I. (1994a). The influence of decision aids on choice strategies under conditions of high cognitive load. *IEEE Transactions on Systems, Man and Cybernetics*, *24*(4), 537-547.

Todd, P., & Benbasat, I. (1994b). The influence of decision aids on choice strategies: An experimental analysis of the role of cognitive effort. *Organizational Behavior and Human Decision Processes*, *60*(1), 36-74.

Todd, P., & Benbasat, I. (1999). Evaluating the impact of DSS, cognitive effort, and incentives on strategy selection. *Information Systems Research*, *10*(4), 356-374.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, *185*(4157), 1124-1131.

Van der Heijden, H. (2004). User acceptance of hedonic information system. *MIS Quarterly*, *28*(4), 695-704

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, *46*(2), 186–204

Vessey, I. (1985). Expertise in debugging computer programs: A process analysis. *International Journal of Man-Machine Studies*, *23*(5), 459-494.

Vessey, I. (1991). Cognitive fit: A theory-based analysis of the graph versus tables literature. *Decision Sciences*, *22*(2), 219-240.

Vessey, I., & Conger, S. (1993). Learning to specify requirements: The relationship between application and methodology. *Journal of Management Information Systems*, *10*(2), 171-201.

Vessey, I., & Galletta, D. (1991). Cognitive fit: An empirical study of information acquisition. *Information Systems Research*, *2*(1), 63-84.

Vessey, I., & Weber, R. (1986). Structured tools and conditional logic: An empirical investigation. *Communications of the ACM*, *29*(1), 48-57.

Vitalari, N. P. (1985). Knowledge as basis of expertise in systems analysis: An empirical study. *MIS Quarterly*, *9*(3), 221-241.

Waltz, D., Elam, J., & Curtis, B. (1993). Inside a software design team: Knowledge acquisition, sharing, and integration. *Communications of the ACM*, *36*(10), 63-77.

Wand, Y., & Weber, R. (1990). An ontological model of an information system. *IEEE Transactions on Software Engineering*, *16*(11), 1282-1292.

Wand, Y., & Weber, R. (1993). On the ontological expressiveness of information systems analysis and design grammars. *Journal of Information Systems*, *3*(4), 217-237.

Wand, Y., & Weber, R. (2002). Research commentary: Information systems and conceptual modeling – a research agenda. *Information Systems Research*, *13*(4), 363-376.

Wand, Y., & Weber, R. (2004). Reflection: Ontology in information systems. *Journal of Database Management*, *15*(2), iii-vi.

Wang, S. (1996).Toward formalized object-oriented management information systems analysis. *Journal of Management Information Systems*, *12*(4), 117-141.

Wang, W., & Benbasat, I. (2009). Interactive decision aids for consumer decision making in e-commerce: The influence of perceived strategy restrictiveness. *Management Information Systems Quarterly*, *33*(2), 293-320.

Weber, R. (1996). Are attributes entities? A study of database designer's memory structures. *Information Systems Research*, *7*(2), 137-160.

Weber, R. (2003). Still desperately seeking the IT artifact. *MIS Quarterly*, *27*(2), iii-xi.

Webster, J., & Martocchio, J. J. (1992). Microcomputer playfulness: Development of a measure with workplace implications. *MIS Quarterly*, *16*(2), 201-226.
Weick, K. E. (1995). What theory is not, theorizing is. *Administrative Science Quarterly*, *40*(3), 385-390.
Weick, K. E., & Roberts, K. H. (1993). Collective mind in organizations: Heedful interrelating on flight decks. *Administrative Science Quarterly*, *38*(3), 357-381.
Weinberg, G. (1971). *The psychology of computer programming*. New York, NY: Van Nostrand Reinhold.
Wixom, B. H., & Ariyachandra, T. (2011). State of business intelligence in academia 2010. *BI Congress II.* Retrieved from http://www2.commerce.virginia.edu/bicongress/content/2010-State-of-BI-Report.pdf
Ye, L. R., & Johnson, P. E. (1995). The Impact of explanation facilities on user acceptance of expert system advice. *MIS Quarterly*, *19*(2), 157-172.
Yoon, Y., Guimaraes, T., & O'Neal, Q. (1995). Exploring factors associated with expert system success. *MIS Quarterly*, *19*(1), 83-106.
Zachary, W. W. (1988). Decision support systems: Designing to extend the cognitive limits. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 997-1030). Amsterdam, Holland: Elsevier.
Zhang, P. (2000). The effects of animation on information seeking performance on the World Wide Web: Securing attention or interfering with primary tasks? *Journal of the Association for Information Systems*, *1*(1), 1-28.
Zhang, P., Li, N., Scialdone, M. J., & Carey, J. (2009). The intellectual advancement of human-computer interaction research: A critical assessment of the MIS literature (1990-2008). *AIS Transactions on Human-Computer Interaction*, *1*(3), 55-107.
Zigurs, I., & B. K. Buckland (1998). A theory of task/technology fit and group support systems effectiveness. *MIS Quarterly*, *22*(3), 313-334.
Zmud, R., Anthony, W., & Stair, R. (1993). The use of mental imagery to facilitate information identification in requirements analysis. *Journal of Management Information Systems*, *9*(4), 175-191.
Zmud, R., & Lim, J. (2011). *The nature and development of common ground: Explaining performance variation across cognitively-diverse groups* (Working paper).

About the Authors

Michael J. DAVERN (Ph.D., University of Minnesota) is Professor of Accounting and Business Information systems in the Department of Accounting at the University of Melbourne where he is also director of the Master of Business & IT program. His primary research focus is on IT-enabled decision making in business contexts such as risk management, revenue management, and transaction processing. In 2001, he co-founded the IS-CORE, the AIS special interest group on cognitive research, and was inaugural president. His prior work has appeared in *Decision Support Systems, Communications of the ACM, Journal of Management Information Systems*, and *Information Technology and People*, among others. His work is supported by the Australian Research Council (LP100100068), He is currently an AE at the AIS transactions on HCI, a past program co-chair of the Australasian Conference on Information Systems, and a past doctoral fellow and faculty mentor at the ICIS Doctoral Consortium.

Teresa M. SHAFT is Associate Professor of Management Information Systems at The University of Oklahoma's Michael F. Price College of Business. She received her Ph.D. in Management Information Systems from the Pennsylvania State University. Her research interests focus on the cognitive processes of systems developers, IT effectiveness and the role of information systems in environmental management. Her research appears in journals including *Journal of the Association for Information Systems, MIS Quarterly, Information Systems Research, Journal of Management Information Systems,* and *Business and Society*. She is a co-founder of IS-CORE, a special interest group of AIS devoted to cognitive research in IS. Her research has been supported through grants from the U.S. National Science Foundation.

Dov TE'ENI holds the Mexico Chair for Information Systems at Tel Aviv University. He studies how computers support people deciding, communicating, sharing knowledge and interacting. His research usually combines model building, laboratory experiments and development of prototypes like *Spider* and *kMail*. Dov has published over 100 academic papers with over 70 colleagues. He has co-authored with Jane Carey and Ping Zhang *Human-computer interaction for developing effective organizational systems* (Wiley) and co-edited with David Schwartz the *Encyclopedia of Knowledge Management*. He is currently the President of AIS – the international Association of Information Systems, has served as Senior Editor for *MIS Quarterly*, *AIS Transactions on HCI* and associate editor for *Journal of AIS, Information and Organizations*, and *Internet Research*. He is co-editor of *European Journal of IS*. Dov was awarded *AIS Fellowship* in 2008.