Task Complexity and Informing Science: A Synthesis

T. Grandon Gill
University of South Florida
Tampa, Florida, USA

ggill@coba.usf.edu

Richard C. Hicks
Texas A&M International University
Laredo, Texas, USA

rhicks@tamiu.edu

Abstract

Task complexity is a construct widely used in the behavioral sciences to explore and predict the relationship between task characteristics and information processing. Because the creation and use of IT in the performance of tasks is a central area of informing science (IS) research, it follows that better understanding of task complexity should be of great potential benefit to IS researchers and practitioners. Unfortunately, applying task complexity to IS is difficult because no complete, consistent definition exists. Furthermore, the most commonly adopted definition, objective task complexity, tends to be of limited use in situations where discretion or learning is present, or where information technology (IT) is available to assist the task performer. These limitations prove to be severe in many common IS situations.

The paper presents a literature review identifying thirteen distinct definitions of task complexity, then synthesizes these into a new five-class framework, referred to as the Comprehensive Task Complexity Classes (CTCC). It then shows the potential relevance of the CTCC to IS, focusing on different ways it could be applied throughout a hypothetical information systems lifecycle. In the course of doing so, the paper also illustrates how the interaction between different classes of task complexity can serve as a rich source of questions for future investigations.

Keywords: task complexity, systems development, objective complexity, task structure, software complexity, information processing

Introduction

> *The driving force behind the creation of informing environments and delivery systems is that a task needs to be accomplished. (Cohen, 1999)*

The concept of a task is central to the Informing Sciences (IS). As the driving force in the development of informing systems, the role played by task is of particular significance to the individuals responsible for developing such systems, since better insights into the task to be accomplished should improve their understanding of the resources that will be required. In this context, task complexity appears to be a particularly important characteristic. The construct is widely used in the behavioral sciences to explore the relationship between task characteristics and cognitive activities. It is hypothesized to be:

Editor: Eli Cohen

1. A determinant of the information processing and cognitive load that will be required in order to perform a task (Benbasat & Todd, 1996; Campbell, 1988).
2. An objective basis for determining compensation for a task (e.g., Auster, 1989).
3. A key determinant of task performer intrinsic motivation, satisfaction and goal acceptance (e.g, Beer, 1968; Jimmieson & Terry, 1999; Seybolt, 1976; Wood, Mento & Locke, 1987)
4. An important determinant of the general and specific knowledge required in order to perform a task (Wood, 1986)
5. An important determinant of appropriate training method for a given task (Bolt, Killough, & Koh, 2001)
6. A critical factor in the selection of decision-making strategies and information seeking behaviors (e.g., March & Simon, 1958; Payne, 1976; Vakkari, 1999)
7. A variable impacting the appropriate method for displaying information in order to achieve effective decision-making (Speier & Morris, 2003)
8. A factor determining the value of data quality information in data warehousing situations (Fisher, Chengalur-Smith, & Ballou, 2003)

Given that the coming of the "information age" is being accompanied by a transition from physical work to "'knowledge work" (Drucker, 1989), and that discontinuous change (Handy, 1990) and high-velocity environments (Bourgeois & Eisenhardt, 1988) are leading to dramatic changes in the nature of tasks and jobs being performed, it is reasonable to anticipate that the potential value of being able to predict how abstract task characteristics will influence the intended task-completion system (Cohen, 1999) should grow correspondingly.

Given both task complexity's importance in the behavioral literature and its integral relationship to information processing activities, it is somewhat surprising that the construct has not been used more extensively in the informing sciences. With few exceptions (e.g., Handzic, 2001), most of the research in the area has been situated in the field of management information systems (MIS). Even there, its use has been limited to a rapidly growing a number of studies that have used the construct as an independent variable (e.g., Barki, Rivard, & Talbot, 1993; Blili, Raymond, & Rivard, 1998; Bolt, et al., 2003; Fisher, et al., 2003; Gill, 1996; Jacko & Salvendi, 1996; Kishore, Agrawal, & Rao, 2004; Liao & Palvia, 2000; Mascha, 2001; Mykytyn & Green, 1992; Roberts, Cheney, Sweeney, & Hightower, 2004; Speier & Morris, 2003; Swink & Speier, 1999;) in the prediction of some other outcome. There are also examples of research that employ MIS-related tasks as part of general investigations of the construct (e.g., Campbell & Gingrich, 1986). What is notably absent from the IS literature is a systematic examination of the nature of the construct. As the present paper will demonstrate, such an examination is long overdue—as it is hard to imagine any other construct could equal task complexity in terms of the level of ambiguity and internal inconsistency achieved over the years.

Applying task complexity to IS situations has been made even more difficult by the way the construct has commonly come to be defined. Over the past two decades, the most widely used definition in the behavioral literature treats the construct as a function of objective task characteristics. Defined in this way, task complexity is most relevant to static, well-understood tasks (as will be explained later in the present paper). Such a limitation of task domain, however, dramatically reduces the construct's potential applicability to the IS field. As IS researchers and practitioners, we routinely encounter situations where the tasks we consider are neither static nor fully understood. Three important examples are:

1. *Development of applications to perform tasks.* How does the complexity of a task impact the design, construction and implementation of information technologies appropriate for performing that task?
2. *Transformation of tasks by IT*. How does the complexity of a task change after it has been transformed through the adoption of an information system?
3. *Shared tasks*. How does the complexity of a task change when the performance of the task is shared between a human performer and an information system?

We believe that the question "how can task complexity be applied to IS?" is an important one. The present paper proposes an organizing framework intended to clarify the applicability of task complexity to IS theory and practice. We begin by reviewing the many existing definitions of task complexity that have been proposed in the behavioral literature. In the course of doing so, we also draw parallels to analogous constructs that have been proposed in software engineering and the cognitive sciences. These definitions, thirteen in all, are then organized into a taxonomy consisting of five classes of task complexity that we have developed. Finally, we discuss the specific application of the complexity classes to a number of IS-related situations, and show how the interaction between different classes of task complexity can serve as a rich source of experimental hypotheses. We believe that this research can serve as an important first step in making the task complexity construct more applicable to IS research and practice.

Nature of a Task

Prior to addressing task complexity, it is useful to present definitions for three terms that are used throughout the present paper: task, problem space, and discretion. All three prove to be critical in understanding the nature of task complexity.

Task

At least four distinct theoretical frameworks for studying tasks have been proposed (Hackman, 1969), treating tasks 1) in stimulus-response terms ("Task qua task"), 2) as a set of required behaviors, 3) as a set of resultant behaviors, and 4) as a set of abilities requirements (Wood, 1986). The present paper does not make any judgments regarding the relative suitability of these different approaches to defining a task. Indeed, we believe consistent task complexity definitions can be established using any of the task conceptions, so long as they are not mixed. As a starting point, the present paper adopts a definition of task that has been widely used in the past (Hackman, 1969):

> *Definition*: a task is a set of assigned a) goals to be achieved, b) instructions to be performed, or c) a mix of the two.

By defining task in this manner, we allow for tasks to be framed either in stimulus-response terms or as required behaviors. We also specifically seek to separate task and task-context, the latter including broader factors such as the physical and social setting in which the task is performed. Later in the paper, we return to this issue by considering how context factors, such as external demands on the task performer, can be expected to impact various conceptions of task complexity.

Problem Space

As already noted, abilities requirements are sometimes used as an alternative way of defining task. Such requirements are central to some task complexity definitions, but to avoid confusion we prefer to use another term to refer to them. Specifically, we adopt the term problem space

(Newell & Simon, 1972), used extensively in the field of cognitive science. We define it as follows:

> *Definition:* A problem space is a representation of the cognitive system that will be used to perform a task "described in terms of (1) a set of states of knowledge, (2) operators for changing one state into another, (3) constraints on applying operators, and (4) control knowledge for deciding what knowledge to apply next" (Card, Moran, & Newell, 1983; p. 87).

Within a problem space, knowledge is often classified in two forms: programs and meta-knowledge. Programs represent domain-specific knowledge that is normally the result of experience or training in the specific task to be performed, or very similar tasks. Meta-knowledge, in contrast, consists of knowledge about knowledge. It typically provides the task performer with access to a variety of search intensive, general problem solving strategies, such as reasoning by analogy (Rumelhart & Norman, 1981) and decomposition (Simon, 1981). These strategies (sometimes referred to as "weak methods"—so named because they offer no guarantees of success and are often quite inefficient) allow task performance to continue in the absence of programs.

Discretion

Under the definition of task that we have chosen, two assignments—1) serve cake for dessert, and 2) perform the steps in the recipe (for baking a chocolate cake) found on page 675 of The Joy of Cooking (Rombauer & Becker, 1975) commencing at 4:30 PM today—would both be valid tasks. To distinguish between the two tasks, it is useful to define a final term:

> *Definition:* the ability of a task performer to choose and/or sequence the activities associated with performing a given task is referred to as task discretion.

Using this definition, the first set of instructions (i.e., "serve cake for dessert") offers high levels of discretion, while the second offers nearly none.

Another way of viewing discretion is in terms of the set of acceptable problem spaces that are invoked by the task stimulus. For a low discretion task, relatively few problem spaces will meet the task requirements. For a high discretion task, on the other hand, many different problem spaces may exist that meet task requirements. As a result, rising discretion will typically be accompanied by a rise in the variety of task performance behaviors that are observed across different performers.

A Task Model

The relationship of task, problem space and discretion can be illustrated using a simple model, presented in Figure 1:

- For tasks with little or no discretion, the task stimulus invokes a specific problem space, which is then applied during task performance.
- For tasks with high discretion, the task stimulus invokes a collection of problem spaces that all meet, or may meet, task requirements. From that collection, a specific space is selected, which is then applied during task performance.

For some high-discretion tasks, weak methods may need to be employed because no invoked problem space offers an acceptable and fully determined path to task completion. In this case, a typical strategy is to use decomposition to break the task into subgoals, then perform each separately. This process adds an iterative component to task performance, illustrated by the dotted arrow in Figure 1. Such tasks are often referred to as ill-structured.

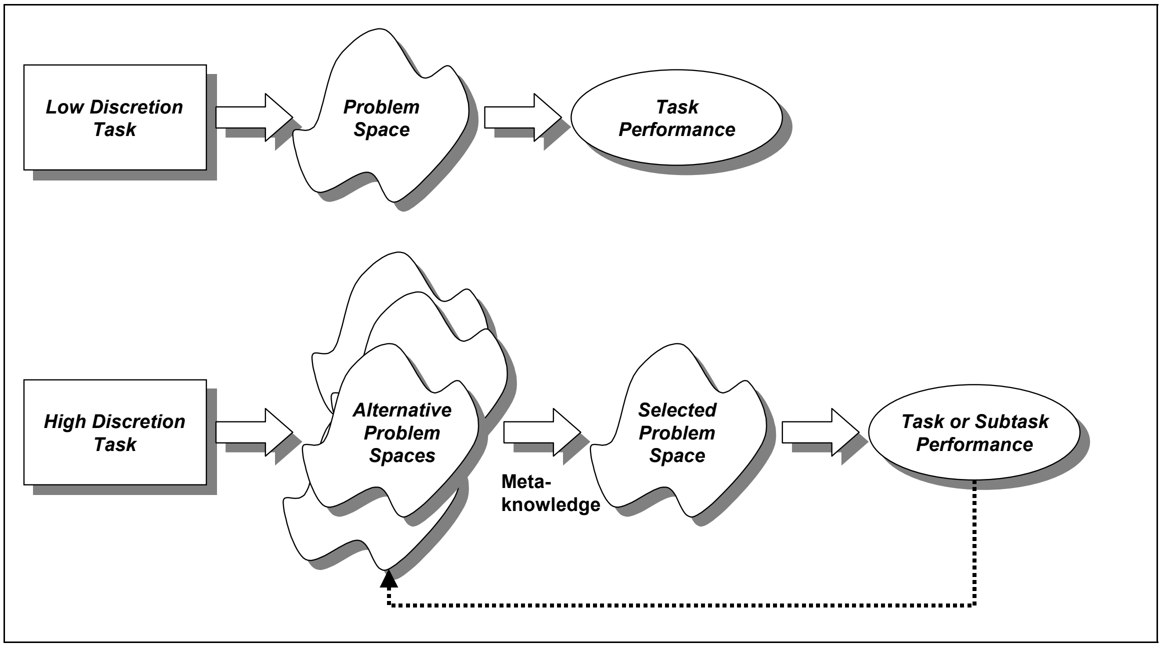

Figure 1: Relationship of Task, Problem Space and Discretion

Task Complexity: A Review

In this section, we review existing definitions of task complexity. We begin by discussing the most widely used class of definitions—objective complexity. We then present a broader survey of the construct as it has been applied in the behavioral literature.

Objective Complexity

Within the behavioral literature, two definitions of task complexity are most widely cited, those proposed by Wood (1986) and Campbell (1988). The first definition (Wood, 1986) is based on the premise that the only task complexity definitions likely to exhibit construct validity are those in which the construct is solely a function of the task itself. Referred to as objective task complexity, the definition proposes that task complexity derives from three primary sources: 1) the number of different components associated with the task (component complexity), 2) the level of interaction between the components (coordinative complexity) and 3) the degree to which the relationship between task-related input and output cues changes over time (dynamic complexity). Total task complexity is further proposed to be a weighted sum (or more complex function) of the three objective complexity sources.

The second definition (Campbell, 1988) similarly favors defining task complexity based upon objective task characteristics. The characteristics proposed are: 1) multiple paths, 2) multiple end states, 3) conflicting interdependence, and 4) uncertainty or probabilistic linkages. While the two definitions seem quite different, there is a strong underlying common theme:

> *Definition:* objective complexity defines task complexity as a function whose value depends strictly upon the characteristics of the task to be performed.

Unfortunately, objective complexity definitions have limited potential applicability to many common IT-related situations. Throughout the present paper, we focus on three such situations:

1) the application of task complexity to systems development.

2) the application of task complexity to tasks transformed by IT, and

3) the application of task complexity to situations where task performance is shared between human performers and IT.

Relating these to the IS framework (Cohen, 1999), the first situation—systems development—can be characterized as an intent to change to the informing environment by creating a system using well understood designs or by creating a system of a totally new design. The relevant questions then become: what does the complexity of the task to be accomplished tell us about the changes we'll have to make to the informing environment and what are the implications for the delivery environment? The last two situations specifically relate to the nature of the task completion system. The relevant questions here become: what, if any, impact on task complexity occurs when IT is introduced to into a task completion system and can better understanding of task complexity provide us with useful insights into the nature of that system? We now consider the inherent limitations of objective complexity when applied to these situations.

Task complexity and systems development

The development of information systems to perform a given task is an area that is at the heart of the informing sciences, the creation of the delivery system (Cohen, 1999). It is also, arguably, the area in which objective task complexity (or concepts very similar to it) has been most extensively applied. Indeed, a typical system specification represents an almost ideal foundation from which objective complexity measures (e.g., number of functional components, component interdependence) can be estimated. Moreover, systems development has also been used as a domain for investigations into objective task complexity (Campbell & Gingrich, 1986).

The first issue that needs to be addressed in applying objective complexity to systems is determining the proper task domain. Two plausible alternatives exist: 1) we might want to estimate the complexity of the task of developing the system or 2) we might want to estimate the complexity of the task to be performed by the system. Given a full specification, a reasonable case can be made that measures of objective complexity computed on the basis of the specification represent the predicted complexity of the development task to be performed (Tran, Lévesque, & Meunier, 2004). This use of objective complexity comes with important limitations, however. The first limitation is that a full design specification must be in place. While specification will normally precede development in specification-based models, such as the waterfall model (Sommerville, 1996), many evolutionary development programming approaches, such as agile methods, permit major changes to design and specifications to merge incrementally during the development process. The second limitation is one of practicality: the value of being able to predict the complexity of the development process is much greater at the outset of the process than in the middle of the process. Unfortunately, by the time a full set of specifications and design documents are completed and signed off in a typical waterfall model development project, developers are often nearing the midpoint of the project.

Even assuming a full specification is in place, there is little basis for treating that specification as a proxy for the objective complexity of the task to be performed by the system. For non-trivial systems, such specifications are rarely "determined" by the task itself. Instead, they evolve only as decisions regarding representation, approach and delivery platform are made. Thus, while it might be valid to measure the objective complexity *of the development task* implied by a particular system design, it would not be valid to assert that such complexity reflects the "objective" complexity of the *task to be performed by the system*. Referring back to Figure 1, systems development tends to be a high-discretion task where the process of moving from task to final problem space (i.e., full specification) is central to task performance.

Task complexity and tasks transformed by IT

Objective complexity can also be very difficult to apply in situations where tasks are transformed by IT. A recurrent theme in MIS research is that IT is dramatically changing the way we work (e.g., Applegate, Cash, & Mills, 1988). The ability to apply a task complexity construct to a continuous stream of new tasks would therefore be of particular value to the MIS field. In practice, however, the objective attributes that will ultimately prove to be relevant to a new task often need to be discovered (Jaikumar & Bohn, 1986). To measure objective complexity, however, these attributes need to be identified and understood. As a result, in a highly dynamic task environment, the objective complexity of new tasks is likely to be immeasurable during the precise period when the construct would be of the greatest value.

Task complexity and shared tasks

Another serious limitation of objective complexity is evident when tools are used in performing a task. The voluntary adoption of an information system in performing a task —such as a decision support system (DSS)—would not impact the task's objective complexity, as a matter of definition. (The use of the term "voluntary" is important here, as a new requirement to use a tool represents a change to the task itself.) The problem that this creates is that use of a tool can dramatically impact many aspects of task performance that the presence of task complexity is supposed to predict (e.g., cognitive demands of the task; Campbell, 1988). Thus, availability of discretion in the use of tools can strip objective complexity of nearly all of its predictive power.

Alternative Task Complexity Definitions

Although objective complexity is often considered in the context of MIS-related tasks (e.g., Bolt, et al., 2001), limitations to its applicability—such as those just presented—significantly reduce its potential value in the IS field. Objective complexity, however, is only one form of task complexity present in the behavioral literature. We now turn to a more systematic review and synthesis of a broader range of complexity definitions, gathered according to the following procedure:

1. The task complexity research cited in earlier reviews (Wood, 1986; Campbell, 1988), either jointly or individually, was examined.
2. Usages of task complexity subsequent to the earlier reviews were identified using the ABI/Inform database, and a further gathering and analysis of included references was performed.
3. Selected references applying constructs related to task complexity in non-administrative disciplines (e.g., cognitive science, information theory, computer science) were identified as a means of broadening the present paper's perspective.

Three distinct issues had to be addressed as part of the review: 1) research that did not specifically identify how the construct was defined, 2) research where the definition and the operationalization were inconsistent, and 3) research where multiple definitions were proposed. In the first case, a definition consistent with usage was assumed. In the second and third cases, each definition was cataloged separately. Upon completing this process, we were able to identify thirteen relatively distinct definitions of task complexity, summarized in Table 1.

Table 1: Existing Task Complexity Constructs

Construct Type	Description	Examples (References may fall in more than one category)
1. Degree of difficulty	Definition treats task complexity as a measure of the task's potential for being perceived as difficult by the task performer. May be operationalized based upon performer-reported assessments of difficulty, or upon indirect measures, such as the degree to which the task must be constantly attended to.	Blili, et al. (1998); Huber (1985); Nordqvist, Hovmark, & Zika-Viktorsson (2004); O'Donnell, Koch, & Boone (2005); Ursic & Helgeson (1990); Wofford, Goodwin, & Premack (1992)
2. Sum of JCI or JDS factors	Defines task complexity in terms of the task's potential to induce a state of arousal or enrichment in the task performer, operationalized using instruments such as the JCI (Job Characteristics Index) or JDS (Job Diagnostic Survey).	Koszowski & Hults (1986); Nordqvist, et al. (2004); Schanke, Bushardt, & Spottswood (1984); Specht (1986)
3. Degree of stimulation	Definition treats task complexity as a measure of the task's potential to induce a state of stimulation or arousal in the task performer. Similar to degree of difficulty except that it is normally measured using physiological measurements (e.g., pupil dilation) as opposed to self-reporting.	Driver & Streufert (1966); Gardner (1990); Kreitler, Zigler, E. & Kreitler (1974)
4. Amount of work required to complete the task or information load associated with the task	Definition treats task complexity as a measure of a task's potential to induce various information processing levels, such as peak processing rate (e.g., bits/second) or total amount of processing (e.g., bits processed). Such processing is intended to be measured objectively, instead of being based on task performer perceptions or responses. It is also nearly always constructed so that task performance strategy is held constant.	Asare & McDaniel (1996); Barki, et al. (1993); Barrow (1976); Benbasat & Todd (1996); Campbell & Gingrich (1986); Coll, Coll, & Thakur (1994); Earley (1985); Gilliland & Landis (1992), Ho & Weigelt (1996), Roberts, et al. (2004); Schweizer (1996), Seybolt (1976); Speier, Vessey, & Valacich (2003)
5. Amount of knowledge	Definition is based upon the amount of knowledge that must be acquired in order to perform the task. Definition may be operationalized using metrics such as the amount of time required to learn the task.	Ackerman (1992); Gill (1996); Wood (1986)
6. Size	Defines task complexity using the information theory of the minimum theoretical size of the problem space necessary to perform the task. Most commonly used in assessing software complexity.	Li & Vitanyi (1993)

7. Number of paths	Defines task complexity in terms of the number of alternative paths that are possible using given performance strategy. Has been used for both general tasks and for analyzing the complexity of computer programs, where it measures the number of branches (e.g., if constructs).	Barki, et al. (1993); Campbell (1988), Jacko & Salvendy (1996); McCabe (1976),
8. Degree of task structure	Defines task complexity as the degree to which a task is not programmed, i.e., the degree to which accepted task-specific procedures for performing the task do not exist. Lack of structure can result from a number of sources, including the lack of a clear goal state to be achieved, the inability to establish the initial starting point of the task and relevant task attributes, and/or a lack of knowledge of strategies suitable for moving between the initial state and the goal state.	Abdolmohammadi & Wright (1987); Barki, et al. (1993); Bronner (1986); Bystrom & Jarvelin (1995); Collins & Hull (1986); Gilliland & Landis (1992); Jimmieson & Terry (1999); Lengnick-Hall & Futterman (1985); Mascha (2001); Nordqvist, et al. (2004); Sheer & Chen (2004); Smith (1988); Vakkari (1998,1999); Zemelman, Di Marco, Norton (1985),
9. Non-routineness or novelty of task	Defines task complexity in terms of the degree to which the task is unfamiliar to the task performer. A routine task is typically viewed as the opposite of a complex task under this definition.	Beer (1968); Frew (1981), Jehn, Northcraft, & Neale (1999); Jiambalvo & Pratt (1982), Jimmieson & Terry (1999); Schwartzwald, Koslowsky, M. & Ochana-Levin (2004); Wagner & Gooding (1987)
10. Degree of uncertainty	Defines task complexity as the degree to which actual performance of the task cannot be predicted at the outset of the task owing to uncertainty. Normally, such uncertainty can arise as a result of lack of structure (see above) or from stochastic uncertainties inherent in the task itself.	Barki, et al. (1993); Belardo & Pazer (1985); Kishore, et al. (2004); Taylor (1987), Te'eni (1989); Wagner & Gooding (1987)
11. Complexity of underlying system or environment	Definition specifically relates to the task of controlling or predicting the behavior of systems. Defines task complexity in terms of the objective attributes of the system (e.g., number of components, degree of interrelationships).	Culnan (1983); Dorner & Scholkopf (1991); Funke (1991); Kottlemann & Remus (1989)
12. Function of alternatives and attributes	Specifically focused on choice tasks, definition treats task complexity to be an objective function of the alternatives available in the task (e.g., number of alternatives, discriminability) and the task's attributes (e.g., the number of criteria needing to be considered, the degree to which they are interdependent).	Fisher, et al. (2003); Javalgi (1988); Klemz & Gruca (2003); Lussier & Olshavsky (1979); Olshavsky (1979); Paquette & Kida (1988); Payne (1976); Pollay (1970); Swink & Speier (1999),.

13. Function of task characteristics	A more general version of the attributes-alternatives definition of task complexity, it defines task complexity to be direct function of all possible task characteristics, such as inherent uncertainties in the nature of the task, tradeoffs that must be made between different goal criteria, and the degree to which steps taken in performing the task are irreversible.	Argote, Insko, Yovetich, & Romero (1995); Auster (1989); Belardo & Pazer (1985); Bolt, et al. (2001); Campbell (1988); Chesney & Locke (1991); Earley (1985); Fisher (1993); Jones, Hesterly, & Borgatti (1997); Klein, Wesson, Hollenbeck, Wright, & DeShon (2001); Korsgaard & Diddams (1996); Liao & Palvia (2000); Mykytyn and Green (1992); Roberts, et al. (2004); Speier & Morris (2003), Speier, et al. (2003), Tran, et al. (2004); Weingart (1992); White & Lui (2005); Wood (1986), Wood, Bandura, & Bailey (1990); Wood, Mento, & Locke (1987)

In organizing these thirteen definitions, we initially attempted to group them into three previously proposed perspectives (Campbell, 1988). These perspectives are:

1. The psychological experience perspective, defining task complexity in terms of its impact on task performers (e.g., perceived difficulty).
2. The task-person interaction perspective, defining task complexity as a construct that can only be determined when both the task and the individual performing the task are taken into account.
3. The objective characteristics perspective, defining task complexity in a manner consistent with objective complexity, as previously discussed.

A number of definitions we identified fell readily into the first group. In three of these definitions, task complexity was defined to be the underlying source of some observable performer-experienced outcome, such as performer-perceived difficulty (definition 1), enrichment (definition 2), and arousal (definition 3). A fourth definition, information processing (IP) activity (definition 4), was similar except that the outcome—IP activity—could be extended to non-human performers (e.g., processing cycles, bits processed).

At the other extreme, there was a cluster of objective complexity definitions (i.e., definitions 11-13). While the specific task attributes viewed as relevant varied considerably across these definitions, all viewed the complexity of the task to be independent of performer and task-context issues that were not part of the task itself.

The middle group of definitions proved to be the most difficult to classify. With the psychological experience group, they shared a tendency to define task complexity in terms of variables that were, at least in part, likely to be a function of the performer rather than the specific task (e.g., amount of knowledge, task structure, routineness). On the other hand, similar to the objective characteristics group, they treated task complexity as an observable variable in its own right, and not merely as the underlying source of some measurable task-related outcome.

Our analysis of these definitions suggested that a number of problems arise from lumping them together in a single, catchall person-task perspective. For example, many of the definitions within the group were only weakly related to each other, such as the novelty of a task (definition 9) and the number of paths available for task performance (definition 7). In addition, the use of the term “person” in "person-task" appeared unduly restrictive, especially if we are interested in person-IT pairings. Furthermore, a number of useful ways of thinking about task complexity have been developed in the field of computer science (e.g., Kolmogorov complexity (Li & Vitanyi, 1993); cyclomatic complexity (McCabe, 1976)), where the task performer is not a person. Finally, some of the definitions could move between perspective categories depending upon how a task is presented. For example, where a task consists solely of prescribed activities (i.e., programs), both uncertainty (definition 10) and number of paths (definition 7) are likely to be objective characteristics of the task itself (Campbell, 1988). On the other hand, where a task is presented purely in terms of goals, and discretion is present, these same definitions can be highly dependent upon the problem space utilized by the task performer. For example, if we have two task experts who choose to perform a given task in different ways, can we really state that the number of paths experienced by each expert is an objective attribute of the task itself?

The middle group of definitions also proves to be particularly important because many involve concepts that have demonstrated relevance to MIS development and implementation. For example:

- Size (definition 6), lack of structure (definition 8) and inexperience with technology (definition 9) have been found to be the primary sources of development risk in large IT projects (Cash, McFarlan, & McKenney, 1988),
- Number of paths (definition 7) per unit of software has been found to be a predictor of software defects (e.g., Gill & Kemerer, 1991).
- Lack of structure (definition 8) has been identified as an important task characteristic in choosing between types of IT (e.g., conventional system, DSS and expert system; Luconi, Malone, & Scott Morton, 1986).

To derive an internally consistent classification, we extended the three original perspectives into five classes of complexity definitions that, taken together, seemed to encompass all widely used definitions that we identified in the literature.

Five Classes of Task Complexity

In rethinking the three task complexity perspectives, we found that existing definitions could be organized using two dimensions: performer dependence and area of focus.

Performer Dependence

The performer dependence dimension deals with the question of whether or not the defined complexity of a task necessarily depends on the nature of the performer. Five of the Table 1 definitions are inherently performer dependent. The first three of these—(1) degree of difficulty, (2) sum of JCI or JDS factors, (3) degree of stimulation—are strongly influenced by a broad range of non-task factors, including attitudes and personal capabilities. Two additional definitions—(8) degree of task structure, and (9) non-routineness or novelty of task—are likely to be influenced both by experience performing the task and by the meta-knowledge that the performer has accumulated since birth. One could, in theory, overcome these dependencies by applying these constructs only to low-discretion, practiced tasks—both eliminating the need for meta-knowledge and controlling for experience. Doing so would be pointless, however, since it would guarantee that the resulting tasks would be characterized as both structured and routine—making them "low complexity" by definition.

Seven definitions, in contrast, are specifically constructed so that they are not necessarily performer dependent. One of these definitions—(4) amount of work required to complete the task or information load associated with the task using specific strategy—is nearly always applied in situations where task performance strategy is controlled (i.e., it is measured for a particular problem space). Three of the definitions—(5) amount of knowledge, (6) size, (7) number of paths—

are metrics of a specific problem space, which could (at least in theory) be employed by different performers. The last three definitions—(11) complexity of underlying system or environment, (12) function of alternatives and attributes, and (13) function of task characteristics—are forms of objective complexity, specifically limited to task attributes.

Only one definition, degree of uncertainty (10), cannot readily be classified along the performer dependence dimension. The problem here is the ambiguity in its usage present in the literature. Sometimes, "uncertainty" is used to describe the inability to identify the most appropriate available strategy during performance of the task owing to lack of task performer knowledge. In such cases, uncertainty can be classified as performer dependent, and is closely related to lack of structure. On other occasions, however, the term is used to describe probabilistic linkages inherent in the task (e.g., Campbell, 1988), such as outcomes in a game that depend upon the roll of a die. In such cases, uncertainty is performer independent.

Area of Focus

The second dimension used in classifying complexity definitions is area of focus. Task complexity definitions generally emphasize one or more of three areas: 1) task complexity as the underlying source of some measurable outcome during task performance (e.g., information processing activity), 2) task complexity as a descriptor of the characteristics of the internal process to be used to perform the task (e.g., the problem space) and 3) task complexity as a function of some collection of inputs that are determined by the task itself (e.g., number of attributes and alternatives).

Of the thirteen definitions, four fall into the outcome category: (1) degree of difficulty, (2) sum of JCI or JDS factors, (3) degree of stimulation, and (4) amount of work required to complete the task or information load associated with the task. In these definitions, presence of the specified outcome is de facto proof that task complexity is present. Another three fall into the internal process category: (5) amount of knowledge, (6) size, and (7) number of paths. In these definitions, task complexity cannot be determined without knowing quantitative and/or qualitative information regarding how the task is to be performed. Three of the remaining definitions define complexity as a function of inputs that are strictly determined by the task itself: (11) complexity of underlying system or environment, (12) function of alternatives and attributes, and (13) function of task characteristics.

A final set of definitions, (8) degree of task structure, (9) non-routineness or novelty of task, and (10) task uncertainty, are best described as constructs that depend on the interaction between the inputs and internal problem space areas of focus. For example, a given task may have some cases (i.e., combinations of inputs) that are routine and some that are novel for a given performer. The degree of structure present in a task, and associated levels of uncertainty associated with task performance, can also be sensitive to inputs. A good example of this, drawn from the MIS area, is the problem of scalability, occurring where a well-defined approach to a task involving a small number of inputs (e.g., low objective complexity) ceases to be effective for larger versions of the same task. In artificial intelligence, the frequent failure of "toy problem" systems to grow into practical intelligent systems is well documented (Dreyfus, 1981).

Five Task Complexity Classes

The thirteen complexity definitions, organized according to the performer dependence and area of focus dimensions, are presented in Table 2.

Table 2: Definitions Organized by Performer Dependence and Area of Focus

		Area of Focus		
		Outcomes	*Internal*	*Inputs*
Performer Dependence	*Dependent*	1. Degree of difficulty 2. Sum of JCI or JDS factors 3. Degree of stimulation	8. Degree of task structure 9. Non-routineness or novelty of task 10. Degree of uncertainty (present during task performance)	
	Independent	4. Amount of work required to complete the task or information load associated with the task (using a specified strategy).	5. Amount of knowledge 6. Size 7. Number of paths	10. Degree of uncertainty (inherent to task) 11. Complexity of underlying system or environment 12. Function of alternatives and attributes 13. Function of task characteristics

Based upon this analysis, we concluded that existing complexity definitions fall into five classes, collectively referred to as the comprehensive task complexity classes (CTCC):

1. *Experienced complexity*: definitions where complexity is defined in terms of what the task performer experiences.
2. *Information processing complexity*: definitions where complexity is defined to be the underlying source of IP capacity requirements or throughput experienced while performing the task.
3. *Problem space complexity*: definitions where complexity is measured as a function of the characteristics of a problems space used to perform the task.
4. *Lack of structure complexity*: definitions where complexity represents the degree to which a task is fully programmed.
5. *Objective complexity*: definitions where complexity is measured as a function dependent upon characteristics strictly specified by the task itself.

The area of focus for each of the five classes of complexity, presented in the context of our task model, is illustrated in Figure 2.

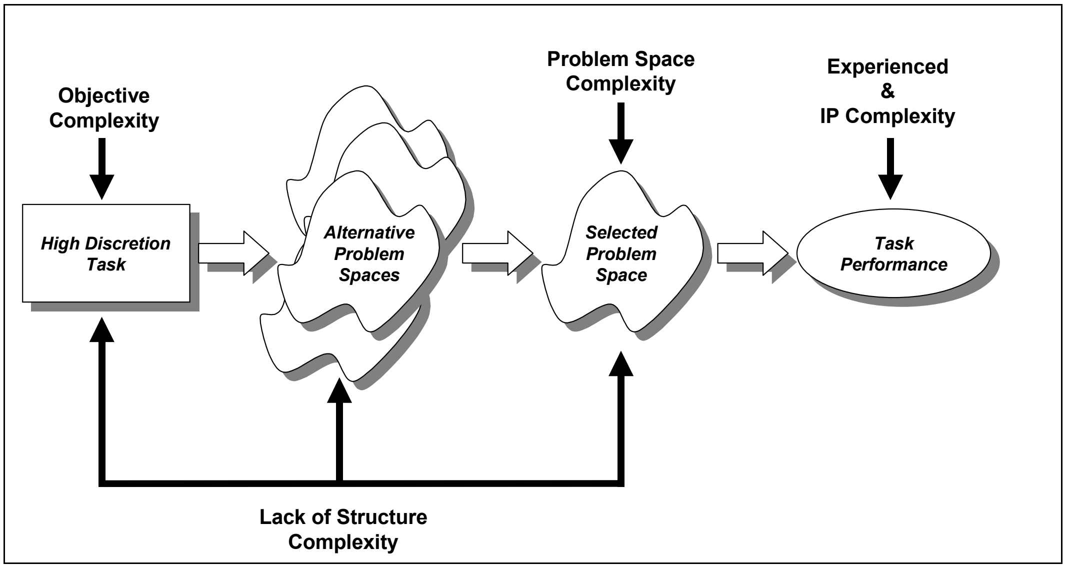

Figure 2: Complexity Classes and Area of Focus

Five Questions

To support the existence of the CTCC, some analysis of construct validity and discriminability is warranted. In this context, we propose that five questions can be particularly helpful:

A. Does the definition explicitly require that high task complexity be accompanied by high levels of information processing?

B. Does the definition directly relate the complexity of a task to the amount of, or qualitative nature of, knowledge employed in performing the task?

C. Does the definition allow task complexity to be influenced by factors external to the task?

D. Does the definition treat task complexity as a construct that can be meaningfully determined for a single instance of a specific task?

E. Does the definition allow the level of complexity to vary according to the approach chosen by the task performer in completing the task?

Each of these questions is now considered.

Question A

> *Does the definition explicitly require that high task complexity be accompanied by high levels of information processing?*

Some relationship between IP requirements (e.g., information load and cues to be attended to; Steinmann, 1976) and task complexity is expected to exist in most of the definitions examined. In the case of definition 4, the relationship is effectively tautological: IP activity is the visible manifestation of underlying task complexity. In other words, task complexity is defined as the task's potential to induce IP activity. Such IP activity has, in turn, been measured in a number of ways. Cumulative measures (e.g., total bits processed), corresponding to the total amount of "work" required to perform the task have been proposed (e.g., Barrow, 1976; p. 435), as have measurements based upon minimum processing capacity requirements (e.g., Schweizer, 1996; p. 116). It has also been observed that IP activity is typically highly correlated with experienced IP measures (Bronner, 1986), implying experienced measures (definitions 1-3) should generally track similarly constructed IP measures (definition 4).

Many researchers treat the various forms of objective complexity (definitions 11-13) as being the underlying source of IP activity (e.g., Campbell, 1988, p. 43). Such a relationship, however, is best viewed as hypothetical rather than definitional. The presence of task discretion can complicate the relationship considerably. A well-established task complexity research stream (e.g., Handsic, 2001; Payne, 1976; Payne, Bettman & Johnson, 1993) demonstrates how increasing sources of objective complexity (e.g., number of alternatives and attributes) can lead decision-makers to adopt strategies that conserve processing resources. In other words, they simplify the task to make it more manageable. Thus, where discretion is present, increases in objective complexity can, paradoxically, lead to reduced information processing.

For the remaining six definitions (definitions 5-10), there is no obvious relationship between task complexity and information processing activity. For those definitions dealing with task performer knowledge and the characteristics of the strategies available to the performer (definitions 5-7), it is nearly impossible to state, in any general way, whether or not tasks requiring a lot of knowledge will also require a lot of information processing when they are performed. Some probably will, some undoubtedly won't. Even where relationships have been proposed, they are ambiguous (e.g., multiple paths may lead to more information processing or less information processing;

Campbell, 1988, p. 43). For those definitions based upon how well the task is understood and specified prior to performance (8-10), the tradeoff between preparation and deliberation (Newell, 1991) implies that—in the absence of adequate task-specific knowledge—cognitively intensive general search strategies (i.e., weak methods) may have to be employed by the task performer. Weighed against this, low structure tasks are inherently high in discretion. As a result, strategies that conserve cognitive resources may be available, such as ignoring all but a few available alternatives.

Question B

Does the definition directly relate the complexity of a task to the amount of, or qualitative nature of, knowledge employed in performing the task?

The relationship between task knowledge and task complexity definitions is one of the most interesting questions that we identified in our review. Some researchers propose that task complexity is a key determinant of the amount of knowledge required to perform a task (e.g., Wood, 1986), and therefore expect task knowledge and task complexity to be positively correlated—particularly for the objective characteristics definitions (definitions 11-13). Others would argue that lack of knowledge is the principal source of lack of task structure, meaning that task complexity should drop as knowledge accumulates (definitions 8-10), implying a negative correlation between lack of structure and problem space classes. Drawing upon research in automatism (e.g., Shiffrin & Dumais, 1981) and on the development of programs (e.g., March & Simon, 1958), similar arguments can be made that experienced complexity (definitions 1-3) is likely to decline as knowledge increases.

A particularly interesting subset of the definitions propose that task complexity can, itself, be defined in terms of the amount of knowledge required to perform the task (e.g., Ackerman, 1992; Gill, 1996) or in terms of qualitative characteristics of the strategies available for performing the task (e.g., number of paths; Campbell, 1988). While these types of definitions (definitions 5-7) do not appear with great frequency in the administrative behavioral literature, they are central to attempts to quantify the complexity of software. For example, in information theory, the complexity of an object X is "the quantity of information needed for the recovery of an object X from scratch" (Li & Vitanyi, 1993). In task terms, a definition could be proposed that the “complexity of a given problem space is the smallest amount of information necessary to recreate that problem space”. An alternative definition could be proposed based upon computing the number of possible paths though a particular problem space (i.e., determining all possible sequences of inputs and the resulting behaviors of the problem space). Such a measure would closely correspond to the cyclomatic complexity metric developed for software (McCabe, 1976)

Definitions 5-7 all make the implicit assumption that task complexity should be roughly equivalent across different task performers who are employing identical or very similar problem spaces (e.g., as the result of similar training). In the IS context, it also leaves open the possibility that the complexity of an automated system that performs a task by applying approaches comparable to those employed by the human performer—e.g., an expert system (Freedman, 1987)—could be treated as equivalent to the complexity of the task performed by a human. Such equivalence could, in turn, provide a practical means of estimating problem space task complexity, such as the theoretical minimum size of an expert system performing a task using a knowledge base that mimics the problem solving approach of a particular human task performer.

Question C

Does the definition allow task complexity to be influenced by factors external to the task?

As already noted, a strong correlation between the experienced complexity (definitions 1-3) and IP complexity (definition 4) measures is predicted in the literature. Unlike most of the other complexity classes, however, experienced complexity measures may also be influenced by stimuli external to the task, such as task context. Cognitive, motivational and social qualities can all lead to changes in the perception of complexity, even when the underlying information processing requirements are unchanged (Bronner, 1986). In other words, we may experience a task as being more complex when we are simultaneously subjected to external stresses that are not part of the task itself. Thus, external demands can hypothetically distort the normal relationship between perceived and actual IP, causing a given level of IP activity to be perceived in different ways.

External factors can also influence complexity based on lack of structure definitions (8-10). The interaction of external demands and meta-knowledge regarding the demands of different task performance strategies may lead to the adoption of strategies that change the degree of task structure. For example, to conserve cognitive resources when external demands are high, a manager might choose to purchase the first acceptable alternative presented by a vendor when making an unfamiliar decision, rather than engaging in general problem solving activities (e.g., extensive research)—thereby transforming a low-structure situation to a high-structure situation. Alternatively, when external task demands are low (or if the performer's need for cognitive stimulation is high) the performer might choose to try a new approach to performing a familiar task, thereby transforming the routine to the non-routine. As a secondary effect, such changes could also have a dramatic impact on experienced and actual complexity, to be addressed more fully in the discussions of Question E.

For the remaining classes of complexity (i.e., problem space complexity and objective complexity), factors external to the task are irrelevant to the constructs.

Question D

> *Does the definition treat task complexity as a construct that can be meaningfully determined for a single instance of a specific task?*

Another very useful question for distinguishing between the task complexity definitions deals with whether or not a definition is applicable to a single instance of a task, or can only be determined meaningfully for a collection of task instances. The distinction between overall task complexity and individual task case complexity has been made, for example, in examining complexity changes brought about by the adoption of expert systems (Gill, 1996).

For most of the definitions proposed in the literature, it is reasonable to assign separate complexity levels to individual task instances. For example, the process of diagnosing and treating a patient with the common cold could be characterized as being less complex than diagnosing a treating a patient with pneumonia (e.g., there are more tests to run and a wider range of possible treatments). Similarly, it could be argued that the task of diagnosing a common cold is roughly the same complexity when performed by either doctors or nurse-practitioners, since they are both likely to ask the same questions and prescribe the same treatment (i.e., “take two aspirin, go to bed, and call me in the morning”).

The problem space characteristics definitions (definitions 5-7) present an entirely different situation with respect to the task instance vs. collection issue. To continue the above illustration, the problem space task complexity of the diagnosis task for the above-mentioned doctor and nurse practitioner are likely to be quite different. When we go to a doctor, we do so believing that the task performer has the knowledge necessary to diagnose and treat a much broader range of conditions than we would if we went to a nurse. We do not go to a doctor because we know we have the common cold, we go to a doctor (in preference to a nurse) because we are afraid we might have something more serious. Thus, characterizing a task as “diagnosing the common cold” is usually not meaningful from a problem space perspective—except, perhaps, in a triage situation where the sole assignment given to the task performer is to divide patients up into "common cold" and "not common cold" groups. The actual task is "medical diagnosis", and the problem space complexity of that task (e.g., size of the problem space, number of paths) derives from the number and variety of possible cases that the task performer's problem space is expected to address—not from the activities that occur or knowledge applied during the performance of a single task instance. The task of playing chess is another example. We would expect the problem space complexity of the chess-playing task to be a function of some expected level of performance (e.g., the set of opponents the player could be expected to beat some threshold percentage of the time), not of the ability to play a "legal" game of chess.

A good example of the practical ramifications of the task instance vs. collection issue was discovered during the construction of early medical expert systems. The MYCIN system was able to diagnose blood infections at a level exceeding that of most specialists (Jackson, 1986). The Achilles heel of the system was that it could only achieve such a high success rate for task instances that were, in fact, blood infections. It failed miserably on any case that had similar indications but was not, in fact, caused by a blood infection (a problem sometimes referred to as "knowledge fragility" (Waterman, 1986)). Unfortunately, in a practical clinical setting, it is usually impossible to separate applicable and non-applicable cases in the early stages of diagnosis. Thus, a problem space based purely on a very narrowly defined set of cases often proves to be of limited practical value.

Question E

> *Does the definition allow the level of complexity to vary based on the approach chosen by the task performer in completing the task?*

In the event task discretion is present, the impact of the performer's strategy choices on task complexity can be quite different across definitions. As an illustration, suppose that it was Friday afternoon and that the owner of a small business needed to choose between two detailed (and superficially similar) vendor proposals for installing a new computer network in his or her company—a decision that needed to be made at once. Suppose, as well, that the owner had a tee time at the country club in an hour. Under such circumstances, the owner might come up with two alternate strategies for choosing between the proposals: comparing the proposals line by line (thereby missing the golf game) or flipping a coin. Would the owner's decision to flip a coin be treated as reduction in the complexity of the task?

For the psychological experience and information processing definitions (definitions 1-3 and 4, respectively), the answer would clearly be yes—the decision reduced the cognitive demands of the task. We have already presented arguments that such a choice also changes the degree of task structure (definitions 8-10), as a coin flip is highly structured whereas the task of evaluating proposals for a new network is likely to be non-routine (for most managers). For the remaining definitions, however, the performer's decision to flip a coin would not impact task complexity. The choice of a particular strategy does not impact the nature of a particular problem space (definitions 5-7). Similarly, the decision to flip a coin does not, in any way, change the task characteristics that are the source of objective complexity (definitions 11-13).

Summary

The matrix of questions and definitions is presented in Table 3. As made apparent by the pattern of responses to the questions posed, the five proposed classes, at a minimum, appear to be necessary for organizing the existing definitions. The pattern of responses also suggests that attempts to define more than five classes could lead to discriminability problems.

Table 3: Summary of Definitions versus Key Questions

	#	*Definition of construct*	*A. IP Activity*[a]	*B. Nature or Quantity of Task Knowledge*[a]	*C. Influenced by External Task Context*[b]	*D. Applicable to Individual Task*[b]	*E. Affected by Choice of Strategy*[b]
I. Experienced by Performer	1	Degree of difficulty	Correlated (+)	Correlated (-)	Yes[c]	Yes	Yes
	2	Sum of JCI or JDS factors	Correlated (+)	Correlated (-)	Yes[c]	Yes	Yes
	3	Degree of stimulation provided by task	Correlated (+)	Correlated (-)	Yes[c]	Yes	Yes
II. Information Processing Requirements	4	Amount of work required to complete the task or information load associated with performing the task (using a specified strategy).	Yes	No	No	Yes	Yes
III. Problem Space	5	Amount of knowledge	No	Yes	No	No	No
	6	Size of problem space	No	Yes	No	No	No
	7	Number of paths	Correlated (+/-)	Yes	No	No	No
IV. Lack of Structure	8	Non-routineness or novelty of task	No	Correlated (-)	Yes	Yes	Yes
	9	Degree of task structure	No	Correlated (-)	Yes	Yes	Yes
	10	Degree of uncertainty	No	Correlated (-)	Yes	Yes	Yes
V. Objective Characteristics	11	Complexity of underlying system	Correlated (+/-)	Correlated (+)	No	Yes	No
	12	Function of alternatives and attributes	Correlated (+/-)	Correlated (+)	No	Yes	No
	13	Function of task characteristics	Correlated (+/-)	Correlated (+)	No	Yes	No

[a] *Yes:* The definition assumes that task complexity is defined in terms of the associated characteristic. *No:* The definition does not assume that task complexity is defined in terms of the associated characteristic—nor is there likely to be any strong correlation. *Correlated:* Although task complexity is not defined in terms of the associated characteristic, the expectation is that a strong correlation between the characteristic and task complexity will be observed. Such correlation may be + (i.e., factor is expected is expected to vary directly with task complexity), - (i.e., factor is expected to vary inversely with task complexity) or +/- (i.e., a relationship between the factor and task complexity is likely to exist, but is not necessarily predicted to be monotonic. For example, rising complexity may lead to increased information processing that may, ultimately, trigger the performer's use of simplifying strategies—resulting in reduced information processing).

[b] *Yes:* the associated characteristic is applicable to the definition. *No:* the associated characteristic is not directly applicable to the definition.

[c] A secondary effect may result from the result of changing strategy (consistent with Question E).

Discussion

In this section, we consider how the comprehensive task complexity classes might be applied to the three areas of specific interest to IS that were previously identified, i.e., systems development, task transformation and task sharing. In these discussions, potential areas for future research are also identified.

Task Complexity and Systems Development

Task complexity's relevance to systems development can be illustrated in the context of a simplified systems development life cycle (SDLC), such as that presented in Figure 3. As illustrated by the box above the SDLC stages, most existing applications of task complexity in the systems lifecycle occur once design is finalized. From a conceptual standpoint, this makes sense. By establishing a system design—and freezing further design changes, as is recommended in a strict SDLC setting—we dramatically reduce the discretion associated with the remainder of the development task. We would therefore expect measurements of objective complexity to be useful predictors of other complexity classes, such as information processing (e.g., work required to develop the system).

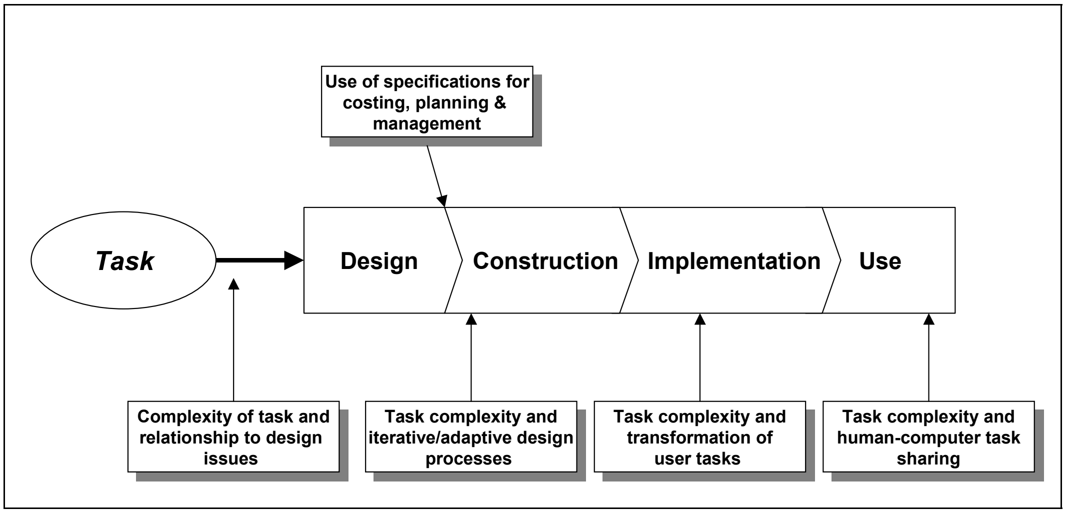
Use of specifications for costing, planning & management
Task
Design
Construction
Implementation
Use
Complexity of task and relationship to design issues
Task complexity and iterative/adaptive design processes
Task complexity and transformation of user tasks
Task complexity and human-computer task sharing

Figure 3: Task Complexity Issues and the SDLC

Figure 3 also identifies examples of important IS topics for which task complexity is used far less extensively. Two of these topics—the task transformation that occurs during IS implementation and the complexity of tasks shared between user and IT—are discussed at greater length later in this section. For now, we focus on the issues most relevant to the early stages (design and construction).

Pre-design Task Complexity

At the very outset of a development project, our "given" is typically the task (or collection of tasks) that we intend to perform using IT, the driving force for informing systems (Cohen, 1999). At this point in the project, the large amount of discretion present implies that objective complexity—by itself—is unlikely to be very useful. From an economic standpoint, however, any insights into the project provided by an understanding of task complexity would be extremely valuable. The design stage of the SDLC often consumes a substantial fraction of a development budget. It can therefore be very costly to wait until design is complete before discovering that the objective complexity of the task-design combination we have finalized implies excessive costs or unacceptable development time.

There is some evidence that understanding the comprehensive complexity of a task may prove useful in the early stages of the SDLC. For example, we have already noted that measures from the problem space complexity class (e.g., size) and lack of structure class (e.g., lack of task structure, lack of experience with relevant technologies) have been found to be major contributors to IT project risk (Cash, et al., 1988). There have also been attempts to examine how the problem space characteristics of the task map to tool choice, particularly in the domain of expert systems (e.g., Luconi, et al. 1986; Prerau, 1985). Given the economic value of pre-design insights into a project, we believe that further research in these areas is warranted.

Task Complexity and Agile Programming

The ongoing evolution in programming practices being experienced by the MIS area also suggests that we should not become over-reliant on the type of objective complexity measures that a full SDLC design specification offers, such as a complete set of function point descriptions. Agile programming methods often lead to the development of systems through a series of iterations that do not involve the creation of a full specification (Highsmith & Cockburn, 2001). Instead, they rely on the nearly continuous creation of a series of working systems, with changes to the "plan" being welcomed instead of being frozen out. In such a project environment, it seems likely that objective complexity class measures will be of limited use by themselves. Instead, we will increasingly need to focus on understanding interactions between objective complexity measures and other complexity classes (e.g., lack of structure, problem space) to gain useful insights into the development process. As the scope of the tasks we intend to perform using IT grows, we expect the value of such insights to grow correspondingly.

Task Complexity and Task Transformation

A task complexity construct that cannot accommodate changing tasks will be of severely limited value in a dynamic task environment. In such an environment, insights into novel and evolving tasks are likely to be far more useful than the ability to measure the complexity of static, routine tasks. Accepting the widely acknowledged potential of IT to transform work as a given, it follows that task complexity's ability to address transformation issues such as learning and changing task performers will significantly impact its potential value the IS community.

The five complexity classes can be expected to exhibit very different behaviors in a dynamic task environment. These behaviors are of interest for two reasons. First, they provide insights into how task complexity can be applied and interpreted. Second, the expected interaction between task complexity classes can serve as a basis of hypotheses suitable for future research. In this section, we consider how task complexity is impacted by changes within the task performer (e.g., learning, practice) and to the task performer (e.g., enabling new individuals to perform a task).

Task complexity and learning

In theoretical terms, learning can be treated as changes to the task performer's problem space. Such changes occur in a number of ways (Rumelhart & Norman, 1981), ranging from incremental improvements in performance (e.g., accretion) to complete transformation of knowledge (e.g., restructuring).

To understand how the classes of complexity vary as the problem space changes, we need to view these changes in the context of a theoretical model of knowledge acquisition. A fairly common theme—among the many models that exist—treats knowledge acquisition as a series of three loosely defined stages (Ackerman, 1992):

- *Cognitive:* Very little immediate knowledge is present, therefore the bulk of task performance relies on search. Because such search will often involve techniques that access knowledge outside of the task domain (e.g., reasoning by analogy), the potential size of the problem space is: a) likely to be very large as it is effectively unbounded by the demands of the task, and b) likely to differ dramatically across performers. Because search is very processing intensive, information processing will also tend to be high.

- *Associative:* With growing expertise, task-specific knowledge is substituted for search and less information is therefore required (Camerer & Johnson, 1991). As a consequence, the problem space becomes bounded and actually shrinks, meaning that problem space complexity declines. Similarly, IP requirements tend to shrink as efficiency improves.
- *Autonomous:* As the expert grows more advanced, additional knowledge about the task continues to be learned, leading to a gradual increase in size for the problem space and the associated problem space class of complexity. The knowledge versus search tradeoff implies that while problem space complexity continues to grow, other classes of complexity (e.g., IP complexity, experienced complexity) are likely to level off, or decline.

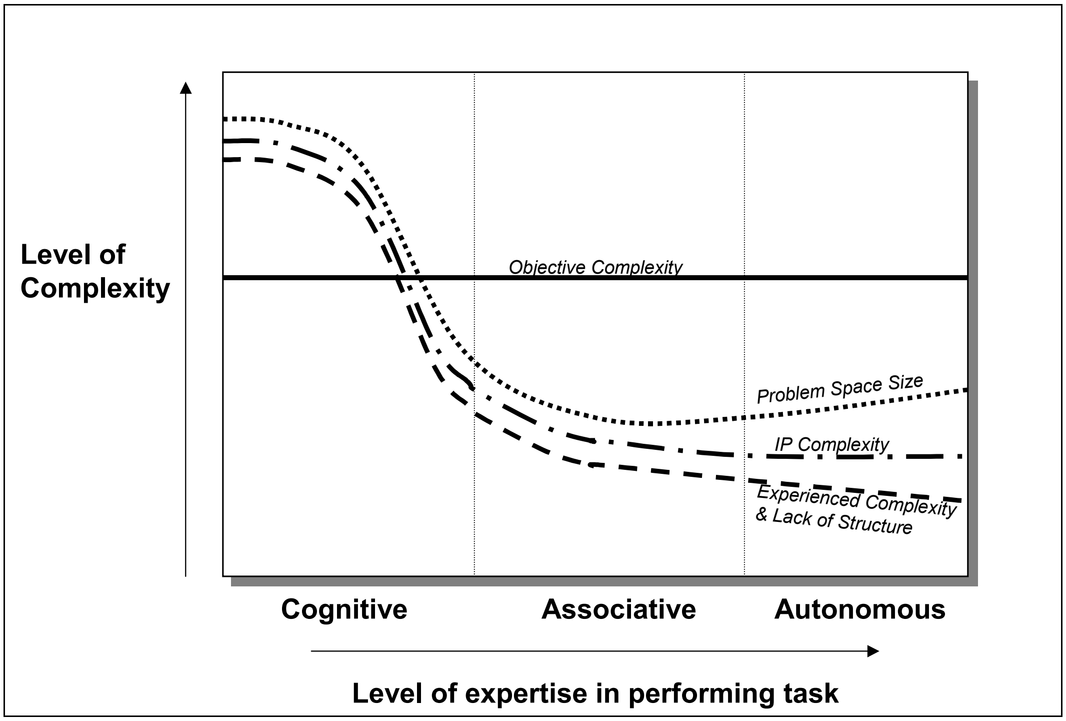

Figure 4: Typical Acquisition of Expertise Model

Progressing through these stages of knowledge acquisition, we would therefore expect to see the following changes in the task complexity classes, illustrated in Figure 4:

- Psychological experience (definitions 1-3) and information processing (definition 4) measures should decline monotonically, as our efficiency in performing the task increases.
- Problem space complexity measures (definitions 5-7) first decline dramatically—as general knowledge is supplanted by task-specific knowledge and more compact problem spaces can be formed. A subsequent gradual decline in problem space size may then occur as knowledge representation becomes more efficient, through schema tuning (Rumelhart and Norman, 1981). Finally, the problem space size begins to rise gradually through accretion. Interestingly enough, the pattern of tuning followed by accretion is also reported in the development of expert systems (e.g., Bachant & McDermott,1984).

- Task structure (definitions 8-10) would continually increase, implying a decline in lack of structure complexity (mirroring that of experienced complexity).
- Objective characteristics measures (definitions 11-13) should remain unchanged throughout the learning process for a given task.

The hypothesized unchanging nature of objective complexity as knowledge is acquired has important implications for its appropriate use. For simple, static task situations—the type most commonly investigated in the behavioral literature—we have already noted that the presence of objective complexity is hypothesized to be a predictor of other forms of complexity, such as information processing (Campbell, 1988) or amount of task knowledge (Wood, 1986). In dynamic task settings, such as those common in IT environments, objective complexity measures are unlikely to be useful for this purpose. On the other hand, objective complexity may prove to be very useful for measuring system capabilities or predicting strategy changes. For example, during the development of R1 (later named XCON) there was a long-term accretion of rules that corresponded to an increase in the number of system types that R1 could be applied to (e.g., Bachant & McDermott,1984). In other words, increasing problem space size was accompanied by increased objective complexity capacity. With respect to task-performer experienced complexity, increasing objective complexity can act as a trigger for changes in task performance strategy that reduce cognitive demands (e.g., Payne, 1976; Payne, Bettman & Johnson, 1993).

Before leaving the subject of learning and task complexity altogether, it is useful to add a comment on the effects of practice on task complexity. As noted earlier, research into the acquisition of expertise (e.g., Shiffrin & Dumais, 1981) has shown that repeated performance of a task leads to automatism, whereby the task's demands on the task performer's cognitive resources will decline significantly. As a consequence, even where task performance strategy is controlled, we would expect psychological experience measures of task complexity (i.e., definitions 1-3) to decline with practice. Assuming, however, that all aspects of task performance are kept constant (e.g., task performance activities are prescribed or the performer problem space does not change), it would be expected that task complexity would not change according to the remaining measures. For example, once a concert performer learns a piano piece well enough to play it from memory, there would be no reason to assume that the complexity of the task of playing the piece declines with further practice—except from the perspective of the performer, who finds that it can be played with less and less conscious attention. Unlike other forms of learning, such practice effects can occur even when the overall structure of the problem space is unchanged.

Task complexity and alternative performers

There are relatively few "real world" tasks where the entire domain of interest is limited to how the task is performed by a particular individual performer. When IT is introduced, the situation often becomes even more convoluted—as such technologies can enable task performance by new individuals (e.g., Gill, 1996). As a consequence, the question of how task complexity is affected by the presence of alternative or multiple performers is highly relevant to IS researchers and practitioners. Conceptually, there are a number of differences that can be present across performers, including:

- *Problem Space:* Different task performers may employ substantially different problem spaces.
- *Practice*: Even where performers employ essentially equivalent problem spaces, practice effects (already discussed) can lead to widely differing cognitive demands

- *Discretion:* Where task discretion is present, even with roughly similar experience levels, choices of individual strategies employed may vary based upon factors not intrinsic to the task, such as accumulated meta-knowledge.

As a result of these differences, task complexity can vary considerably across performers under many of the definitions.

The widest variation in complexity across performers is likely to be for the psychological experience and lack of structure classes (definitions 1-3 and 8-10). For this class, differences in task complexity can be the result of practice, problem space and discretion variations. For the information processing class (definition 4), the practice effect drops out. For the problem space (definitions 5-7) class, only variations in the problem space lead to differences in defined complexity. Finally, for the objective characteristics class (definitions 11-13), we would expect no differences in task complexity across performers.

Task complexity and external tools

In today's information age, a rapidly growing percentage of tasks are, or could be, augmented by the use of IT tools. In such an environment, the relationship between tools and task complexity (in its various forms) needs to be understood. As was the case for alternative performers, the complexity classes differ widely in response to the introduction of tools.

The first issue that must be addressed relating to the use of tools is whether or not a tool is required. The requirement that a tool be employed represents a change in the nature of the task itself. Thus, we would anticipate that changing a task so that use of a tool is required would lead to changes in all the complexity classes.

Where the use of a tool (e.g., a software application) is discretionary, the expected impact on the task complexity classes is much less straightforward. For objective complexity (definitions 11-13), there should be no impact because use of the tool is not specified by the task. As a consequence (similar to what was seen in the case of learning) objective complexity is likely to be of limited use in predicting other complexity measures, such as IP, when tools are available but not required. Also similar to the learning situation, however, objective complexity may prove to be valuable in other contexts—such as estimating the "capacity" of the tool-performer combination or identifying threshold levels for tool use.

For psychological experience (definitions 1-3) and information processing (definition 4) complexity, at the other extreme, we can reasonably suppose that the most common result of voluntary use of an external tool use would be a decline in complexity measures experienced by the human task performer (particularly once the period of learning the tool has passed). It must be emphasized, however, that this expectation applies only to a given task case. Where a task performer must perform a stream of task cases, the technology-performer pairing can impact the mix of task cases performed. An example of such a task-mix change can be found in the case of the Authorizer's Assistant expert system, used to authorize American Express card charges (Gill, 1996). The system reduced the IP requirements associated with any given authorization (e.g., by accessing appropriate databases on different computers and presenting them in a single organized summary to the human authorizer, along with an accept/deny recommendation). However, the system also intercepted most routine approvals and handled them automatically, without user involvement. As a consequence, the task cases that actually reached the human authorizer were, on average, more challenging than the cases that they had previously approved manually.

For problem space (definitions 5-7) and lack of structure (definitions 8-10) complexity, the introduction of a tool can impact task complexity in either direction. On the one hand, the need to learn to use the tool could actually increase the size of the required problem space and could, at least temporarily, reduce task structure. The introduction of word processing technology, for example, probably increased the amount of knowledge required to be an effective typist. On the other hand, where a tool contains embedded expertise (e.g., an expert system), the availability of the tool could reduce the amount of knowledge required to perform a task, and make the task more structured. The introduction of search engines, as another example, dramatically reduced the amount of knowledge required to conduct an effective search of the Internet. In the adoption of expert systems, both increasing and decreasing knowledge requirements resulting from system adoption have been observed (Gill, 1996).

Conclusions

The underlying premise of the present paper is that an understanding of task complexity could offer major benefits within the informing sciences. Unlike some earlier papers on the subject of task complexity, however, we take no position regarding the “correct” definition of the construct. Far too many definitions have been proposed and used to allow us to define it by fiat. Instead, we have directed our attention to establishing five distinct ways that the construct can be consistently and rigorously defined, based on its usage in the literature: 1) as a psychological experience, 2) as a source of information processing potential, 3) as a characteristic of a problem space, 4) as measure of task structure and 5) as a function of task characteristics. We believe this analysis provides a useful point of departure for both researchers interested in applying the constructs and for practitioners seeking to interpret the findings of such research.

We have also drawn a number of conclusions regarding the use the construct within the IS area, including:

- Many IS situations are not well served by limiting task complexity to the "objective complexity" construct that is most commonly applied in the behavioral sciences.
- The potential relevance of different complexity classes may change during the stages of systems development, and may also be strongly impacted by the development approach taken.
- Understanding the behavior of the complexity classes in the presence of learning and human-computer task sharing is of particular interest to IS, since all three elements of the informing sciences framework (Cohen, 1999) come into play: the informing environment, the delivery system and the task-completion system. Considering the interrelationships between the five complexity classes can enhance such understanding, and lead to many useful research questions.

We believe the comprehensive complexity framework we have presented can serve as a useful tool for interpreting past research into task complexity. More importantly, we believe it to be a useful guide for future research that investigates or applies the construct. Such research is likely to be crucial as the scope of the tasks we automate increases, and the way we develop them becomes increasingly adaptive in nature.

References

Abdolmohammadi, M. & Wright, A. (1987). An examination of the effects of experience and task complexity on audit judgments. *Accounting Review, 17*(1), 1-13.

Ackerman, P.L. (1992). Predicting individual differences in complex skill acquisition: Dynamics of ability determinants. *Journal of Applied Psychology*, *77*(5), 598-614.

Applegate, L.M., Cash, J.I. & Mills, D.Q. (1990). Information technology and tomorrow's manager. In W.G. McGowan (Ed.), *Revolution in real time: Managing information technology in the 1990s* (pp. 33-48). Boston: HBS Press.

Argote, L., Insko, C.A, Yovetich, N., & Romero, A.A. (1995). Group learning curves: The effects of turnover and task complexity on group performance. *Journal of Applied Social Psychology*, *25*(6), 512-529.

Asare, S.K. & McDaniel, L.S. (1996). The effects of familiarity with the preparer and task complexity on the effectiveness of the audit review process. *The Accounting Review*, *71*(2), 139-159.

Auster, E.R. (1989). Task characteristics as a bridge between macro- and microlevel research on salary inequality between men and women. *Academy of Management Review*, *14*(2), 173-193.

Bachant, J. & McDermott, J. (1984). R1 revisited: Four years in the trenches. *AI Magazine*, *5*(3), 21-32.

Barki, H., Rivard, S. & Talbot, J. (1993). Towards an assessment of software development risk. *Journal of Management Information Systems*, *10*(2), 203-255.

Barrow, J.C. (1976). Worker performance and task complexity as causal determinants of leader behavior style and flexibility. *Journal of Applied Psychology*, *61*(4), 433-440.

Beer, M. (1968). Needs and need satisfaction among clerical workers in complex and routine jobs. *Personnel Psychology*, *21*, 209-222.

Belardo, S. & Pazer, H.L. (1985). Scope/Complexity: A framework for the classification and analysis of information-decision systems. *Journal of Management Information Systems*, *2*(2), 55-72.

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

Blili, S., Raymond, L. & Rivard, S. (1998). Impact of task uncertainty, end-user involvement, and competence on the success of end-user computing. *Information and Management*, *33*, 137-153.

Bolt, M.A., Killough, L.N., & Koh, H.C. (2001). Testing the interaction effects of task complexity in computer training using the social cognitive model. *Decision Sciences*, *32*(1), 1-20.

Bourgeois, L.J. & Eisenhardt, K.M. (1988). Strategic decision processes in high velocity environments: Four cases in the microcomputer industry. *Management Science*, *34*(7), 816-835.

Bronner, R. (1986). Perception of complexity in decision making processes: Findings of experimental investigations. In Witte & Zimmerman (Eds.), *Empirical Research on Organizational Decision-Making* (pp. 45-64). Amsterdam: North Holland.

Bystrom, K. & Jarvelin, K. (1995). Task complexity affects information seeking and use. *Information Processing and Management*, *31*(2), 191-213.

Camerer, C.F. & Johnson, E.J. (1991). The process-performance paradox in expert judgment: How can experts know so much and predict so badly? In K.A. Ericsson & J. Smith (Eds.), *Towards a general theory of expertise: Prospects and limits* (pp. 195-217). Cambridge: Cambridge University Press.

Campbell, D. J. (1988). Task complexity: A review and analysis. *Academy of Management Review*, *13*(1), 40-52.

Campbell, D.J. & Gingrich, K. (1986). The interactive effects of task complexity and participation on task performance: a field experiment. *Organizational Behavior and Human Decision Processes*, *38*, 162-180.

Card, S.K., Moran, T.P. & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale NJ: Earlbaum.

Cash, Jr. J.I., McFarlan, F.W., & McKenney, J.L. (1988). *Corporate information systems management: The issues facing senior executives* (2nd ed.). Homewood: Irwin.

Chesney, A.A. & Locke, E.A. (1991). Relationships among goal difficulty, business strategies, and performance on a complex management simulation task. *Academy of Management Journal*, *34*(2), 400-424.

Cohen, E. (1999), Reconceptualizing information systems as a field of the transdiscipline informing science: From ugly duckling to swan. *Journal of Computing and Information Technology*, *7*(3), 213-219

Coll, R.A., Coll, J.H. & Thakur, G. (1994). Graphs and tables: A four factor experiment. *Communications of the ACM*, *37*(4), 77-84.

Collins, P.D. & Hull, F. (1986). Technology and span of control: Woodward revisited. *Journal of Management Studies, 23*(2), 143-164.

Culnan, M.J. (1983). Environmental scanning: The effects of task complexity and source accessibility on information gathering behavior. *Decision Science, 14*, 194-206.

Dorner, D. & Scholkopf, J. (1991). Controlling complex systems; or expertise as 'grandmother's knowhow'. In K.A. Ericsson & J. Smith (Eds.). *Towards a general theory of expertise* (pp. 218-239). Cambridge, UK: Cambridge University Press.

Dreyfus, H.L. (1981). From micro-worlds to knowledge representation: AI at an impasse. In J. Haugeland (Ed.). *Mind design* (pp. 161-204). Cambridge: MIT Press.

Driver, M.J. & Streufert, S. (1969). Integrative complexity: An approach to individuals and groups as information processing systems. *Administrative Science Quarterly*, *14*, 272-285.

Drucker, P.F. (1989). *The new realities*. New York: Harper & Row.

Earley, C. (1985). The influence of information, choice and task complexity upon goal acceptance, performance and personal goals. *Journal of Applied Psychology*, *70*, 481-491.

Fisher, C.D. (1993). Boredom at work: A neglected concept. *Human Relations*, *46*(3), 395-417.

Fisher, C.W., Chengalur-Smith, I. & Ballou, D.P. (2003). The impact of experience and time on the use of data quality information in decision making. *Information Systems Research*, *14*(2), 170-188.

Freedman, D.H. (1987). AI meets the corporate mainframe. *Infosystems*, *2*, 32-37.

Frew, D.R. (1981). Diagnosing and dealing with task complexity. *Personnel Administrator*, *26*(11), 87-97.

Funke, J. (1991). Solving complex problems: Exploration and control of complex systems. In Sternberg & Frensch (Eds.). *Complex problem solving: Principles and mechanisms* (pp. 185-222), Hillsdale, NJ: Erlbaum.

Gardner, D.G. (1990). Task complexity effects on non-task-related movements: A test of activation theory. *Organizational Behavior and Human Decision Processes*, *45*, 209-231.

Gill, G.K. & Kemerer, C.F. (1991). Cyclomatic complexity density and software maintenance productivity. *IEEE Transactions on Software Engineering*, *17*(12), 1284-1288.

Gill, T.G. (1996). Expert systems usage: Task change and intrinsic motivation. *MIS Quarterly*, 20(3), 301-329.

Gilliland, S.W. & Landis, R.S. (1992). Quality and quantity goals in a complex decision task: Strategies and outcomes. *Journal of Applied Psychology*, *77*(5), 672-681.

Hackman, J.R. (1969). Toward understanding the role of tasks in behavioral research. *Acta Psychologica 31,* 97-128. North Holland.

Handzic, M. (2001). Does more information lead to better informing?" *Proceedings of Informing Science 2001*. 251-256.

Handy, C. (1990). *The age of unreason*. Boston MA: Harvard Business School Press.

Highsmith, J. & Cockburn, A. (2001). Agile software development: The business of innovation. *IEEE Computer*, September, 120-122.

Ho, T.H. & Weigelt, K. (1996). Task complexity, equilibrium selection, and learning: An experimental study. *Management Science*, *42*(5), 659-679.

Huber, V. (1985). Effects of task difficulty, goal setting and strategy on the performance of a heuristic task. *Journal of Applied Psychology*, *70*, 492-504.

Jacko, J.A. & Salvendi, G. (1996). Hierarchical menu design: Breadth, depth and task complexity. *Perceptual and Motor Skills*, *82*, 1187-1201.

Jackson, P. (1986). *Introduction to expert systems*. Reading: Addison Wesley.

Jaikumar, R. & Bohn, R.E. (1986). The development of intelligent systems for industrial use: A conceptual framework. Research on technological innovation. *Management and Policy*, *3*, 169-211.

Javalgi, R.G. (1988). Influence of task complexity on the predictive performance of the nonlinear and linear disaggregate models of consumer behavior. *Journal of Business Research*, *16*(1), 1-16.

Jiambalvo, J. & Pratt, J. (1982). Task complexity and leadership effectiveness in CPA firms. *Accounting Review*, *57*(4), 734-750.

Jimmieson, N.L. & Terry, D.J. (1999). The moderating role of task characteristics in determining responses to a stressful work simulation. *Journal of Organizational Behavior*, *20*, 709-736.

Jehn, K.A., Northcraft, G.B. & Neale, M.A. (1999). Why differences make a difference: A field study of diversity, conflict and performance in workgroups. *Administrative Science Quarterly*, *44*(4), 741-763.

Jones, C., Hesterly, W.S. & Borgatti, S.P. (1997). A general theory of network governance: Exchange conditions and social mechanisms. *Academy of Management Review*, *22*(4), 911-946.

Kishore, R., Agrawal, M. & Rao, H.R. (2004). Determinants of sourcing during technology growth and maturity: An empirical Study of e-Commerce sourcing. *Journal of Management Information Systems* *21*(3), 47–82.

Klein, H.J., Wesson, M.J., Hollenbeck, J.R., Wright, P.M. & DeShon, R.P. (2001). An assessment of goal commitment: A measurement model meta-analysis. *Organizational Behavior and Human Decision Processes*, *85*(1), 32-55.

Klemz, B.R. & Gruca, T.S. (2003). Dueling or the battle royale? The impact of task complexity on the evaluation of entry threat. *Psychology & Marketing*, *20*(11), 999-1016.

Korsgaard, M.A. & Diddams, M. (1996). The effect of process feedback and task complexity on personal goals, information searching and performance improvement. *Journal of Applied Social Psychology*, *26*(21), 1889-1911.

Kottemann, J.E. & Remus, W.E. (1989). A study of the relationship between decision model naturalness and performance. *MIS Quarterly*, *13*(2), 171-181.

Kozlowski, S.W.J. & Hults, B.M. (1986). Joint moderation of the relation between task complexity and job performance for engineers. *Journal of Applied Psychology*, *71*(2), 196-202.

Kreitler, S., Zigler, E. & Kreitler, H. (1974). The complexity of complexity. *Human Development*, *17*, 54-73.

Lengnick-Hall, C.A. & Futterman, D.H. (1985). Getting a handle on complex units. *Personnel*, *62*(3), 57-63.

Li, M. & Vitanyi, P. (1993). *An introduction to kolmogorov complexity and its applications*. New York, NY: Springer Verlag.

Liao, C. & Palvia, P.C. (2000). The impact of data models and task complexity on end-user performance: An experimental investigation. *International Journal of Human-Computer Studies*, *52*, 831-845.

Luconi, F.L., Malone, T.W. & Scott Morton, M.S. (1986). Expert systems: The next challenge for managers. *Sloan Management Review*. Summer, 3-14.

Lussier, D.A. & Olshavsky, R.W. (1979). Task complexity and contingent processing in brand choice. *Journal of Consumer Research*, *6*, September, 154-165.

March, J.G. & Simon, H.A. (1958). *Organizations*. New York, NY: Wiley & Sons.

Mascha, M.F. (2001). The effect of task complexity and expert system type on the acquisition of procedural knowledge: Some new evidence. *International Journal of Accounting Information Systems*, *2*, 103-124.

McCabe, T.J. (1976). A complexity measure. *IEEE Transactions on Software Engineering*, *SE-2*(4), 308-320.

Mykytyn, Jr., P.P. & Green, G.I. (1992). Effects of computer experience and task complexity on attitudes of managers. *Information & Management*, *23*(5), 263-278.

Newell, A.R. (1991). *Unified theories of cognition*. Cambridge MA: Harvard.

Newell, A.R. & Simon, H.L. (1972). *Human problem solving*. Englewood Cliffs NJ: Prentice Hall.

Nordqvist, S., Hovmark, S. & Zika-Viktorsson, A. (2004). Perceived time pressure and social processes in project teams. *International Journal of Project Management*, *22*, 463–468

O'Donnell, E., Koch, B. & Boone, J. (2005). The influence of domain knowledge and task complexity on tax professionals' compliance recommendations. *Accounting, Organizations and Society*, *30*, 145–165

Olshavsky, R. (1979). Task complexity and contingent processing in decision making: A replication and extension. *Organizational Behavior and Human Performance*, *24*, 300-316.

Paquette, L. & Kida, T. (1988). The effect of decision strategy and task complexity on decision performance. *Organizational Behavior and Human Decision Processes*, *41*, 128-142.

Payne, J.W. (1976). Task complexity and contingent processing in decision making. *Organizational Behavior and Human Performance*, *16*, 366-387.

Payne, J.W., Bettman, J.R. & Johnson, E.J. (1993). *The adaptive decision maker.* Cambridge, UK: Cambridge University Press.

Pollay, R.W. (1970). The structure of executive decisions and decision times. *Administrative Science Quarterly*, *15*, 459-471.

Prerau, D.S. (1985). Selection of an appropriate domain for an expert system. *AI Magazine*, *6*(2), 26-30.

Roberts, T.L., Cheney, P.H., Sweeney, P.D. & Hightower, R.T. (2004). The effects of information technology project complexity on group interaction. *Journal of Management Information Systems*, *21*(3), 223–247.

Rombauer, I.S. & Becker, M.R. (1975). *The joy of cooking*. Indianapolis: Bobbs-Merrill.

Rumelhart, D.E. & Norman, D.A. (1981). Analogical processes in learning. In J.R. Anderson (Ed.). *Cognitive skills and their acquisition* (pp. 335-359). Hillsdale, NJ: Erlbaum.

Schnake, M.E., Bushardt, S.C. & Spottswood, C. (1984). Internal work motivation and intrinsic job satisfaction. *Group and Organization Studies*, *9*(2), 201-219.

Schwarzwald, J., Koslowsky, M. & Ochana-Levin, T. (2004). Usage of and compliance with power tactics in routine versus nonroutine work settings. *Journal of Business and Psychology, 18*(3), 385-402.

Schweizer, K. (1996). The speed-accuracy transition due to task complexity. *Intelligence*, *22*, 115-128.

Seybolt, J.W. (1976). Work satisfaction as a function of the person-environment interaction. *Organizational Behavior and Human Performance*, *17*, 66-75.

Sheer, V.C. & Chen, L. (2004). Improving media richness theory: A study of interaction goals, message valence, and task complexity in manager-subordinate communication. *Management Communication Quarterly : McQ*, *18*(1), 76-93.

Shiffrin, R.M. & Dumais, S.T. (1981). The development of automatism. In J.R. Anderson (Ed.). *Cognitive skills and their acquisition* (pp. 111-140). Hillsdale, NJ: Erlbaum.

Simon, H.A. (1981). *The sciences of the artificial* (2nd ed.). Cambridge MA: MIT Press.

Smith, G.F. (1988). Towards a heuristic theory of problem structuring. *Management Science*, *34*(12), 1489-1506.

Sommerville, I. (1996). Software process models. *ACM Computing Surveys*, *28*(1), 269-271.

Specht, P.H. (1986). Job characteristics as indicants of CBIS data requirements. *MIS Quarterly*, *10*(2), 271-286.

Speier, C. & Morris, M.G. (2003). The influence of query interface design on decision-making performance. MIS Quarterly, *27*(3), 397-423.

Speier, C., Vessey, I. & Valacich, J.S. (2003). The effects of interruptions, task complexity, and information presentation on computer-supported decision making. *Decision Sciences*, *34*(4), 771-797.

Steinmann, D. (1976). The effects of cognitive feedback and task complexity in multiple-cue probability learning. *Organizational Behavior and Human Performance*, *15*, 168-179.

Swink, M. & Speir, C. (1999). Presenting geographic information: Effects of data aggregation, dispersion, and user's spatial orientation. *Decision Sciences*, *30*(1), 169-195.

Taylor, L.A. (1987). Decision quality and commitment within a probabilistic environment. *Organizational Behavior and Human Decision Processes*, *39*, 203-227.

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

Tran, C.D., Lévesque, G., & Meunier, J.G. (2004). Software functional complexity measurement with the task complexity approach. *RIVF'04*, February 2-5, Hanoi, Vietnam, 77-86.

Ursic, M.L. & Helgeson, J.G. (1990). The impact of choice phase and task complexity on consumer decision making. *Journal of Business Research*, *21*, 69-90.

Vakkari, P. (1998). Growth in theories on information seeking: An analysis of growth of a theoretical research program on the relation between task complexity and information seeking. *Information Processing and Management*, *34*(2/3), 361-382.

Vakkari, P. (1999). Task complexity, problem structure and information actions: Integrating studies on information seeking and retrieval. *Information Processing and Management*, *35*, 819-837.

Wagner, J.A. & Gooding, R.Z. (1987). Shared influence and organizational behavior: A meta-analysis of situational variables expected to moderate participation-outcome relationships. *Academy of Management Journal*, *30*(3), 524-541.

Waterman, D.A. (1986). *A guide to expert systems*, Reading: Addison-Wesley.

Weingart, L.R. (1992). The impact of group goals, task component complexity, effort, and planning on group performance. *Journal of Applied Psychology*, *77*(5), 682-693.

White, S. and Lui, S.S. (2005). Distinguishing costs of cooperation and control in alliances. *Strategic Management Journal, 26*, 913–932

Wofford, J.C., Goodwin, V.L. & Premack, S. (1992). Meta-analysis of the antecedents of personal goal level and of the antecedents and consequences of goal commitment. *Journal of Management*, *18*(3), 595-615.

Wood, R. (1986). Task complexity: Definition of the construct. *Organizational Behavior and Human Decision Processes*, *37*, 60-82.

Wood, R., Bandura, A. & Bailey, T. (1990). Mechanisms governing organizational performance in complex decision-making environments. *Organizational Behavior and Human Decision Processes*, *46*, 181-201

Wood, R., Mento, A.J. & Locke, E.A. (1987). Task complexity as a moderator of goal effects: A meta-analysis. *Journal of Applied Psychology*, *72*(3), 416-425.

Zemelman, D., Di Marco, N. & Norton, S.D. (1979). Life style-work group structure-task congruence and job satisfaction and performance: A three dimensional contingency model. *Journal of Management Studies*, *16*(3), 317-333.

Biography

Grandon Gill is an Associate Professor in the Information Systems and Decision Sciences department at the University of South Florida. He holds a doctorate in Management Information Systems from Harvard Business School, where he also received his M.B.A. His principal research areas are the impacts of complexity on decision-making and IS education, and he has published many articles describing how technologies and innovative pedagogies can be combined to increase the effectiveness of teaching across a broad range of IS topics. Currently, he teaches programming, database and managerial courses to both undergraduate and graduate students.

Richard Hicks is an Associate Professor of Information Technology at Texas A&M International University. He holds a Ph.D. from the University of Texas at Austin. His research has appeared in academic journals such as *Decision Support Systems* and *Information and Management*. He is the designer and developer of the artificial intelligence RAD development environment EZ-Xpert, and is currently researching inference without the inference engine.