

# The Effect of the Application Domain in IS Problem Solving: A Theoretical Analysis

Iris Vessey

UQ Business School, University of Queensland

email: [i.vessey@business.uq.edu.au](mailto:i.vessey@business.uq.edu.au)

## Abstract

This study presents theory that formalises, and generalises to problems of different levels of structure, the role of the application domain in IS problem solving. It does so by developing a unifying theory to explain the diverse findings from two experiments that focused on the role of the application domain in IS problem solving. The theoretical framework that we use to form the structure for our theory is a dual-task problem-solving model based on the theory of cognitive fit. Cognitive fit applies to problem solving in each of the contributing domains (application and IS), as well as to the interaction between the two. The theory of cognitive fit allows us to distinguish different types of interactions between the tasks that must be conducted in the IS and application domains when the two types of tasks ‘match’ and when they do not. Those interactions may be supportive, neutral, or conflicting, depending on whether the problem under investigation is well- or ill-structured.

## Introduction

Domain knowledge, which is fundamental to all disciplines (Alexander, 1992), is knowledge of the area to which a set of theoretical concepts is applied. Domain knowledge has long been acknowledged as an important avenue of inquiry in educational research (see, for example, Alexander, 1992; Alexander and Judy, 1988) with studies being conducted in such diverse areas as physics and economics, on the one hand, and history and reading, on the other. Such studies have found that thinking is dominated by content and skills that are domain-specific (McPeck, 1990), and that the lack of domain knowledge results in inelegant problem-solving strategies (Alexander and Judy, 1988).

In the information systems (IS) discipline, the term ‘domain knowledge’ has dual significance. First, *IS domain* knowledge provides representations, methods, techniques, and tools that form the basis for the development of application

systems. Second, those application systems are developed to organise or structure solutions to real-world problems that exist in a given business area, or *application domain*. IS problem solving therefore applies theoretical concepts from the IS domain to the application domain of interest. Hence, knowledge of the IS and the application domains go hand-in-hand in solving IS problems.

A number of studies argue that application domain knowledge impacts IS problem-solving effectiveness (see, for example, Blum, 1989; Curtis et al., 1988; Glass and Vessey, 1992). Few studies have, however, addressed this relationship empirically. Exceptions are Burton-Jones and Weber (1999), Khatri et al (2006), Puroo et al (2002), Shaft and Vessey (1995, 1998, 2006), and Vessey and Conger (1993). Most research has examined processing aspects, with far fewer studies addressing data aspects, such as conceptual modelling. Finally, no studies have presented theory that seeks to explain the role of application domain knowledge in IS problem solving.

Given the pervasiveness of the application domain in IS development, it is important to understand why and how application domain knowledge can aid IS problem solving. In this paper we present theory that explains the roles of both IS and application domain knowledge, and the interactions between the two. We use theory related to dual-task problem solving as the theoretical framework. We then introduce the theory of cognitive fit as the fundamental theory for identifying and explaining different types of interrelationships between IS and application domain knowledge. Finally, we use theory from cognitive psychology in suggesting that the types of interaction between IS and application domain knowledge differ depending on the nature of the problem under investigation. We illustrate our theory with two recently published studies, each addressing different types of problems (Khatri et al., 2006; Shaft and Vessey, 2006).

In the next section, we present the theoretical foundations on which our theory is based. The following two sections present our theory of cognitive fit in dual-task problem solving in well- and ill-structured problems and then we present the implications of our theory for both the IS and cognitive psychology communities. Lastly, we present our conclusions.

## **Theoretical foundations**

In this section, we present three theoretical perspectives that serve as the basis for theory that explains the role of the application domain in IS problem solving. The theoretical framework for establishing roles for both IS and application domain knowledge is provided by research in cognitive psychology that examines problem solvers engaging in the simultaneous solution of two tasks. Formulating problem solving in IS as a dual-task model, that is, with tasks in each of the IS and application domains, allows us to consider situations in which a cognitive

task in one domain has different types of influences on the performance of a cognitive task in the other domain. We present the theory of cognitive fit (Vessey, 1991) as the theoretical basis for determining what happens in such circumstances. Finally, we present theory on the structured nature of the problems under investigation, which we propose as a contingency factor in establishing cognitive fit between the dual tasks.

## Theoretical framework of dual-task problem solving

Following an introduction to the cognitive psychology literature on the simultaneous solution of two tasks, we present a model of dual-task problem solving as a way of thinking about the interrelationship between the two tasks.

### Introducing dual-task problem solving

Cognitive psychologists have long investigated what the community calls 'dual-task interference', a phenomenon that occurs when problem solvers perform two tasks in rapid succession. It is manifested in performance degradation on one or both of the tasks (see, for example, Durso et al., 1998; Koch and Prinz 2002; Navon and Gopher, 1979; Pashler, 1994; Van Selst and Jolicoeur, 1997; Wickens, 2002). When dual-task interference occurs, it is difficult for the individual to allocate attention effectively between tasks, resulting in reduced performance.

Much of the research in the area has focused on the resources needed to conduct the two tasks simultaneously, and therefore the allocation of resources between them (see, for example, Durso and Gronlund, 1999; Kahneman, 1973; Wickens, 2002) and the likelihood of a processing bottleneck (Pashler, 1994; Pashler and O'Brien, 1993; Van Selst and Jolicoeur, 1997). Although there is still substantial debate regarding the mechanisms that underlie the phenomenon, the effects have been observed consistently (see, among others, Koch and Prinz, 2002; Navon, 1990; Navon and Miller, 1987; Pashler, 1994; Pashler and O'Brien, 1993; Van Selst and Jolicoeur, 1997; Whitaker, 1979).

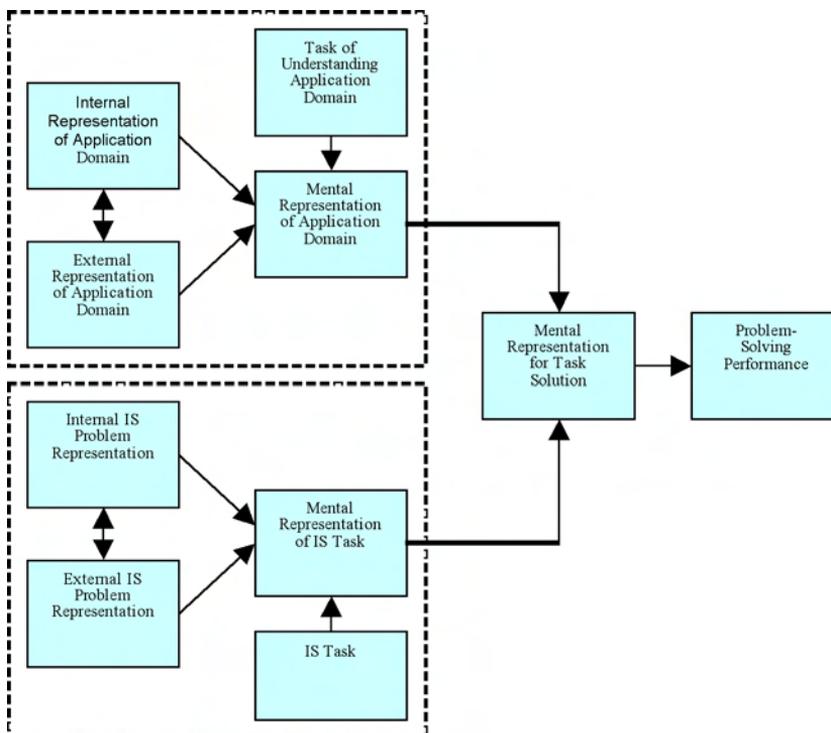
In this research, we apply the basic premises of research on dual-task interference to our specific context of IS problem solving in which interaction occurs between the IS and application domains and therefore between tasks in each of those domains. Under these circumstances, we propose that dual-task problem solving does not always lead to dual-task interference, and we address the circumstances in which the simultaneous solution of the two tasks does lead to dual-task interference and when it does not. We therefore use the term dual-task problem solving in our current analyses, rather than dual-task interference.

### Model of dual-task problem solving in IS

As we have seen, IS problem solving consists of solving problems in a variety of application domains, and therefore knowledge in both the IS and application

domains may play a role in problem solution. The basic premise of our theoretical model is, therefore, that tasks in each of those domains must be solved to reach a solution. We present a dual-task problem-solving model as the framework for examining the interrelationship between the two types of tasks.

Figure 1 presents the dual-task problem-solving model that describes the cognitive process involved in solving a problem in which two types of relevant knowledge interact. This model is based on three repetitions of the basic problem-solving model used to describe cognitive fit (Vessey, 1991), extended to include the notions of distributed cognition proposed in Zhang (1997) and Zhang and Norman (1994). One problem-solving model is used to describe each contributing cognitive task, shown in dashed boxes in Figure 1, with a further model for their interaction.



**Figure 1: Dual-task model of problem solving in an IS context.**

Problem solvers first form mental representations for each contributing task, that is, the cognitive tasks of understanding the application domain (developing a *mental representation of the application domain*) and the IS domain (developing a *mental representation of the IS task*). They must then integrate these two representations into a mental representation that facilitates task solution (the

*mental representation for task solution*).<sup>1</sup> Each contributing task is supported by an *internal problem representation* (knowledge the problem solver has of the domain of interest: IS or application) and an *external problem representation* that presents explicit knowledge related to the solution of the task in the IS domain.

## The theory of cognitive fit

Formulating IS problem solving as a dual-task model opens the way for us to consider situations in which one task might either facilitate or inhibit the other. The theory of cognitive fit that is used as the foundation for the dual-task problem solving model therefore serves as the basis for a theoretical analysis of the circumstances in which application domain knowledge facilitates problem solving, and those in which interaction between the IS and application sub-tasks results in dual-task interference, thereby inhibiting performance.

Here we present the basic notions of the theory of cognitive fit (Vessey, 1991). Although more complex forms of cognitive fit have now been identified (see Vessey, 2006), the theory of cognitive fit is most simply explained in terms of its original formulation as the performance effects resulting from matching the *external IS problem representation* to the *IS task* to be solved (Vessey, 1991). A match or cognitive fit occurs when the information emphasised in a particular *external IS problem representation* matches that required to complete the type of *IS task* under investigation. Decision making is facilitated because the problem-solving processes used to act on the problem representation are similar to those needed to solve the problem.

Using decision making using graphs and tables as our example (see Vessey, 1991) ‘symbolic’ tasks such as determining train departure and arrival times, which involve discrete data values, require the use of analytical processes and are therefore best supported with *external IS problem representations* that also require the use of analytical processes. In this case, such tasks are better supported using tables (symbolic formats) than by graphs. On the other hand, ‘spatial’ tasks such as determining the relationships among the performances of a number of sales regions, which involve making associations or perceiving relationships in the data, require the use of perceptual processes and are therefore best supported with *external IS problem representations* that also require the use of perceptual processes. In this case, such tasks are better supported using graphs (spatial formats) than by tables. Note that when problem-solving processes match, the decision maker is effectively guided in reaching a task solution.

Alternatively, when the type of information emphasised in the *external IS problem representation* does not match that emphasised in the *IS task*, there is nothing to guide the decision maker in working toward task solution, and they must

<sup>1</sup> Note that references to specific constructs in our model are shown in italics.

exert greater cognitive effort to transform the information into a form suitable for solving that particular type of problem (Vessey, 1994). This increased effort will result in decreased performance (that is, decreased decision accuracy, increased decision time, or both).

## Theory on problem structure

We propose that the types of interactions between the tasks in the IS and application domains differ depending on the nature of the problem under consideration. The aspect of the problem that is key in these circumstances is whether it is well- or ill-structured. We distinguish different 'fit' situations based on whether the problem to be solved is well-structured or ill-structured in nature (see Reitman, 1964).

Well-structured problems are those that have a well-defined initial state, a clearly-defined goal state, a well-defined, constrained set of transformation functions to guide the solution process, well-defined evaluation processes, and a single optimal solution path (Greeno, 1978; Sinnott, 1989; Voss and Post, 1988). Further, the information needed to solve the problem is contained in the problem statement.

On the other hand, ill-structured problems are those for which the initial and goal states are vaguely defined or unclear (Voss and Post, 1988), and for which there are multiple solutions and solution paths, or no solution at all (Kitchner, 1983). Further, with such a problem the problem statement does not contain all of the information needed for its solution; hence it is not clear what actions are required to solve it (Chi and Glaser, 1985).

## Dual-task problem solving in well-structured problems

We first present a theoretical analysis of the role of dual-task problem solving and cognitive fit in well-structured problem areas. We then apply the theory to a study of problem solving on the well-structured problem of understanding conceptual schemas (Khatri et al., 2006).

## Implications of problem structure

When the problem is well-structured, both the *external IS problem representation* and the *IS task* itself are sufficiently well formalised for problem solution to occur directly; that is, with reference to the problem statement alone and the associated representations, and without reference to the application domain. In terms of the dual-task problem-solving model presented in Figure 1, problem solving can take place in terms of the cognitive fit model related to the IS task alone (presented at the lower left of the model). In this case, the second task, that of forming a mental representation of the application domain, is not essential to forming the *mental representation for task solution* and therefore plays only a minor role in solving such a problem.

## Role of cognitive fit in dual-task problem solving of well-structured problems

We use the theory of cognitive fit to understand the interrelationship between knowledge of the IS and application domains and the role of each in well-structured problem domains.

Because only IS domain knowledge is required to solve well-structured problems, any effect of application domain knowledge will occur in addition to the effect of IS domain knowledge. There will therefore be no interaction between the two types of knowledge and each therefore has independent effects on performance. We state the following proposition.

- Proposition WS-1: In well-structured IS problem areas, the effects of IS and application domain knowledge are independent.

We can now explore the independent effects of both IS and application domain knowledge on performance. Because IS domain knowledge is essential to solving well-structured problems, we expect that it will influence performance on all types of well-structured problems. We state the following proposition.

- Proposition WS-2: IS domain knowledge influences performance on all tasks in well-structured IS problem areas.

From the viewpoint of the application domain, although application domain knowledge is not essential to the solution of well-structured problems, we expect that its effect will be contingent upon the nature of the task in the well-structured problem area under investigation. Two situations may arise. First, in addressing certain tasks, the knowledge required for task solution can be acquired directly from the *external IS problem representation*; that is, cognitive fit exists. The problem solving that takes place is therefore both accurate and timely (Vessey, 1991). Hence knowledge of the application domain does not influence performance. We state the following proposition.

- Proposition WS-3: When cognitive fit exists, problem solvers addressing tasks in well-structured IS problem areas are equally accurate irrespective of their knowledge of the application domain.

Second, while all of the information essential to solving well-structured problems is available in the *external IS problem representation*, it may not always be available directly. In this case, the knowledge required to address the task and that available for task solution do not match; that is, cognitive fit does not exist. Problem solvers must transform either knowledge emphasised in the *external IS problem representation* to match that emphasised in the *IS task*, or vice versa, in order to form a *mental representation of the IS task* and ultimately a mental representation that facilitates task solution (*mental representation for task solution*). The need to transform such knowledge to solve the task effectively increases

the complexity of the task at hand. In this situation, the presence of application domain knowledge may play a role in problem solution, thereby effectively reducing the complexity of the task under consideration. In terms of the dual-task problem-solving model presented in Figure 1, the formulation of the *mental representation for task solution* may be aided by the presence of application domain knowledge. Hence the cognitive fit model to the upper left of Figure 1 may also play a role in such problem-solving situations. We state the following proposition.

- Proposition WS-4: When cognitive fit does not exist, problem solvers solving tasks in well-structured IS problem areas are more accurate when they have knowledge of the application domain.

## Theoretical analysis of conceptual schema understanding

For our application of dual-task problem solving to well-structured problem areas, we draw on Khatri et al. (2006), who examined conceptual schema understanding in the context of high and low application domain knowledge. Note that, in what follows, we use the practical term ‘schema’ to denote the *external IS problem representation*.

We first address the well-structured nature of conceptual schema understanding, followed by theory on conceptual schema understanding tasks so that we can then examine the situations of fit and lack of fit that may arise.

### The well-structured nature of conceptual schema understanding

A conceptual schema represents the structure and inter-relationships in a set of data. The structure of data has been subject to extensive formalisation over the past four decades (see, among others, Chen, 1976; Codd, 1970; Elmasri and Navathe, 1994). As a result, all of the information required to solve conceptual schema understanding tasks (*IS task*) can be gained from the schema itself, which, from the viewpoint of the model of dual-task problem solving, is represented by the *external IS problem representation*. There is, therefore, a clearly-defined initial state, a well-defined goal state, a formal set of transformation and evaluation paths, as well as a well-defined solution path. Conceptual schema understanding can therefore be addressed using IS domain knowledge alone and we can characterise conceptual schema understanding as a well-structured problem area.

### Characterising conceptual schema understanding tasks

In keeping with the cognitive theories we use to explain the role of application domain knowledge in IS problem solving, we characterise conceptual schema understanding tasks based on the cognitive nature of the task.

Based on prior IS research we can identify two basic types of conceptual schema understanding tasks: comprehension tasks and problem-solving tasks. Comprehension tasks are supported by the education literature, which identifies two different types of knowledge, syntactic and semantic (Shneiderman and Mayer, 1979; Mayer, 1991).<sup>2</sup> We therefore refer to such tasks as *syntactic* and *semantic comprehension tasks*. Syntactic knowledge involves understanding the vocabulary specific to a modelling formalism, for example, the ER model. *Syntactic comprehension tasks* are therefore those that assess the understanding of just the syntax of the formalism (conceptual model) associated with a schema. For example, the syntax for an entity type is a rectangle. Semantic knowledge refers to a set of mappings from a representation language to agreed-upon concepts in the real world. Thus, *semantic comprehension tasks* are those that assess the understanding of the data semantics conveyed through constructs in the schema; for example, a rectangle, the symbol for an entity type, represents a collection of entity instances, that is, objects, things, events, or places (in the 'real world') (Elmasri and Navathe, 1994).

Problem-solving tasks require a deeper level of understanding than comprehension tasks (see Gemino, 1999). Khatri et al. (2006) refer to a problem-solving task that can be solved using knowledge represented *in* the schema as a *schema-based problem-solving task*. Such tasks resemble query tasks; respondents are requested to determine whether, and how, certain information is available from the schema (see also, Shanks et al., 2003). A further type of problem-solving task, which Khatri et al. (2006) refer to as an *inferential problem-solving task*, requires conceptual modellers to use information *beyond* what is provided in the schema (see, for example, Bodart et al., 2001; Burton-Jones and Weber, 1999; Gemino and Wand, 2003; Shanks et al., 2002; Shanks et al., 2003).

In this study, we examined syntactic and semantic comprehension tasks and schema-based problem-solving tasks (in order of increasing complexity) based on their relevance to practicing conceptual modellers.

## Study findings

Khatri et al. (2006) investigated the effects of IS and application domain knowledge on conceptual schema understanding using problem solvers with high and low IS knowledge in both familiar and unfamiliar application domains.

The study findings were as follows. First, there was no interaction between IS and application domain knowledge supporting our theory, as presented in Proposition WS-1, that tasks in well-structured problem areas can be solved

<sup>2</sup> While their work was set in the context of programming languages, these concepts are also relevant to conceptual models.

using IS knowledge alone.<sup>3</sup> Second, as expected, IS domain knowledge influenced the solution of all types of conceptual schema understanding tasks, supporting Proposition WS-2. Third, application domain knowledge did not influence the solution of syntactic and semantic comprehension tasks because the information required for their solution is available directly from the *external IS problem representation*. Hence Proposition WS-3 is supported. The solution of schema-based problem-solving tasks is, however, influenced by the presence of application domain knowledge because the information represented in the schema requires transformation to support the formulation of a consistent *mental representation for task solution*. As we have seen, the presence of application domain knowledge aids in the transformation process, effectively reducing the complexity of these types of tasks. Hence Proposition WS-4 is supported.

## **Dual-task problem solving in ill-structured problems**

In this section, we first present a theoretical analysis of the role of dual-task problem solving and cognitive fit in ill-structured problem areas. We then apply the theory to a study of problem solving on the ill-structured problem of software maintenance (Shaft and Vessey, 2006).

### **Implications of lack of problem structure**

When the IS problem is ill-structured, the problem statement does not contain all of the information required to solve it, and the tasks of developing mental representations of the application and IS domains are both essential to solving the problem. When the knowledge required to solve each of the tasks is consistent (that is, 'fit' exists), then solution performance will be facilitated. When that knowledge is not consistent (that is, 'fit' does not exist), however, there will be a mismatch between the two types of knowledge required for problem solution. In this case, the mental representation of one of the sub-tasks must be transformed to match the other in order to facilitate problem solving. Hence the task is rendered much more complex than when cognitive fit exists.

### **Role of cognitive fit in dual-task problem solving of ill-structured problems**

We again draw on the theory of cognitive fit (Vessey, 1991) to provide the theoretical basis for what happens when knowledge of two tasks is essential to problem solving.

### **Matching representations**

When cognitive fit exists, the problem solver's *mental representation of the application domain* and their *mental representation of the IS task* emphasise similar

<sup>3</sup> Note that this hypothesis was not tested in the original study (Khatri et al., 2006) because conceptual schema understanding was not presented as an example of dual-task problem solving.

types of knowledge and therefore have similar processing requirements. Hence, no transformations are required to form the *mental representation for task solution*, effectively reducing the cognitive resources needed for problem solving (Vessey, 1991). The problem solver can therefore more readily allocate attention between tasks. A problem solver is able to shift attention relatively easily, therefore, between the tasks in the IS and application domains with efforts in the application domain being directed toward the information most relevant to solving the IS task. As a result, *problem-solving performance* is likely to be more accurate and quicker than would otherwise be the case.

We state the following proposition:

- Proposition IS-1: In ill-structured IS problem areas, when the *mental representation of the application domain* is consistent with (i.e., matches) the *mental representation of the IS task*, increased knowledge in the application domain is associated with higher levels of performance on the IS task (*problem-solving performance*).

## Mismatching representations

When cognitive fit does not exist (that is, when knowledge in each of the two task areas does not mutually support problem solving), the problem solver's *mental representation of the application domain* and the *mental representation of the IS task* emphasise different types of knowledge. Without cognitive fit, there is nothing to guide the problem solver in working towards task solution (Vessey, 1991). As a result, activities in the application domain are likely to interfere with the problem solver's ability to complete the IS task, the problem solver may experience difficulty in allocating attention effectively between tasks in the IS and application domains that are not mutually supportive, and performance suffers.

In these circumstances, some kind of transformation must be brought about for problem solving to proceed: either the *mental representation of the application domain* must be transformed to emphasise the same type of knowledge as the *mental representation of the IS task*, or vice versa. The problem solver confronted with these challenges may take one of two approaches to resolve the situation: 1) focus further on the task in the application domain; or 2) focus further on the task in the IS domain.

In the first case, because the *mental representation of the application domain* tends to drive the solution process (Burkhardt et al., 2002), there is a tendency for the problem solver to heed information in the application domain that is consistent with the knowledge already emphasised in the *mental representation of the application domain*, rather than the knowledge consistent with their *mental representation of the IS task* (cf., Broadbent, 1971). Changing the *mental representation of the application domain* to reflect an increased understanding of

knowledge that is consistent with their *mental representation of the IS task* does not, however, enhance the problem solver's ability to complete the *IS task*. Hence, both efficiency and effectiveness are likely to be affected and it is likely that improved knowledge of the application domain will be associated with lower levels of *problem-solving performance*.

In the second case, problem solvers who focus more on task solution perform more effectively than those who focus on the present state (Durso et al., 1998; Hogg et al., 1995; Vessey, 1991). Such an approach would require them to switch attention to their *mental representation of the application domain* only when necessary to resolve an issue directly related to solving the *IS task*. Focusing on the *IS task*, however, has its own difficulties. First, because a problem solver's *mental representation of the application domain* tends to be quite stable over time (Corritore and Wiedenbeck, 1999), problem solvers find it difficult to shift to a different *mental representation of the application domain* after they have invoked an inappropriate one (Taylor et al., 1997). Second, when a problem solver attempts to acquire knowledge consistent with that required to conduct the *IS task* they find difficulty in mapping that knowledge into their mismatched *mental representation of the application domain*. The process of building up the knowledge essential to conducting the *IS task* (that is, developing the *mental representation for task solution*) is therefore quite challenging. Hence, again, both efficiency and effectiveness are likely to be affected and it is likely that improved problem-solving performance will be associated with lower gains in application domain knowledge.

Based on these arguments, therefore, when there is a mismatch between the problem solver's *mental representation of the application domain* and their *mental representation of the IS task*, *problem-solving performance* will be inversely related to improvements in knowledge of the application domain that occur during problem solving. Note that the situation we are describing is that of dual-task interference. We state the following proposition.

- Proposition IS-2: In ill-structured problem areas, when the *mental representation of the application domain* is inconsistent with (i.e., does not match) the *mental representation of the IS task*, greater increases in knowledge of the application domain are associated with lower levels of performance on the *IS task*, and higher levels of performance on the *IS task* are associated with lesser increases in knowledge of the application domain.

In summary, then, when the *mental representation of the application domain* does not support the *mental representation of the IS task* to be conducted:

- paying increased attention to the application domain distracts the programmer from the primary *IS task* of completing the modification; and

- focusing on the *IS task* interferes with the problem solver's understanding of the application domain.

In both cases, the relationship between knowledge of the application domain and performance on the IS task is an inverse one and this is indicative of dual-task interference.

## Theoretical analysis of software maintenance

For our application of dual-task problem solving to ill-structured problems, we draw on Shaft and Vessey (2006). In order to evaluate *problem-solving performance*, the authors used two different types of software maintenance tasks in both familiar and unfamiliar application domains to conduct a study in which the IS task (making a modification to software) and the application domain (represented by the software) were both matched and mismatched.

We first address the ill-structured nature of software maintenance, followed by the nature of both mental representations of the software (*mental representation of the application domain*) and modification tasks (*mental representation of the IS task*). Finally, we present details of the experiment and its findings.

### The ill-structured nature of software maintenance

Software maintenance requires knowledge of both the application area that is being addressed and how to develop as well as maintain software, both of which represent application domain knowledge. Hence application domain knowledge is essential to the solution of ill-structured problems. Allocating attention between tasks in the IS and application domains is particularly relevant to software maintenance because maintainers must comprehend the existing software in order to be able to make relevant changes. Comprehending the software, alone, is a substantive task because maintainers must divide their attention among multiple elements within the software itself (e.g., different modules), as well as understand information other than that in the software (e.g., the external software documentation, etc). Modifying software is also a substantive task, as maintainers must both understand the modification to be conducted as well as make appropriate changes to the software.

Software maintenance is therefore a task for which the initial and goal states are vaguely defined and for which there are no well-defined solution paths. Further, because the problem statement does not contain all of the information needed for solution, it is not clear what actions are required to solve them. We therefore characterise software maintenance as an ill-structured problem area.

## Roles of dual-task interference and cognitive fit in dual-task problem solving

We first examine the intrinsic characteristics of software. We then apply that knowledge to the mental representations that software maintainers form when they examine software (*mental representation of the application domain*), as well as to the types of modification tasks that they may be requested to accomplish (*mental representation of the IS task*).

Part of the essential difficulty of building and maintaining software comes from the need to represent the numerous types of information that are embedded in a piece of software (see Brooks, 1987). We address the characteristics of software and software tasks using these types of information (see, for example, Brooks, 1987; Curtis et al., 1988; Pennington, 1987a, 1987b). The most important types of information embedded in software are function, data flow, control flow, and state information (see, for example, Pennington 1987a, 1987b). Function information reflects the main goals of the program and the hierarchy of sub-goals. Data flow information reflects the series of transformations that data objects undergo. State information relates to the condition-action information embedded in a program; that is, the program actions that result when a set of conditions is true. Control flow information reflects execution sequence; that is, the order in which actions occur.

The mental representations of the software (application domain) formed from these types of information are typically characterised as domain, program, and situation models (von Mayrhauser and Vans, 1995).

First, a software maintainer's *domain model* is a high-level model that focuses on software functionality (Vans et al., 1999). It is more closely aligned to the problem (that is, what the software accomplishes) than how the software accomplishes it (Vans et al., 1999; von Mayrhauser and Vans, 1996) and therefore represents the maintainer's understanding of function information embedded in the software. Second, a software maintainer's *program model* emphasises how the software accomplishes tasks (von Mayrhauser and Vans, 1995) and is quite closely aligned with the programming domain (Pennington, 1987a). It therefore represents the maintainer's understanding of state and control flow information embedded in the software. Control flow reflects the sequencing of actions within software, while state information reflects connections between execution of an action and the state of the software when the action occurs (Pennington, 1987a). Third, a software maintainer's *situation model* allows a maintainer to avoid cognitive overload by reorganising the knowledge gained through detailed study of the software into higher level chunks and reflects the maintainer's understanding of data flow information in the software. It therefore serves as a bridge between the domain and program models (Vans et al., 1999). We can consider the domain

and program models as being at the opposite ends of a continuum, with the situation model residing between them.

When engaging in software comprehension, a software maintainer invokes one of the possible mental representations of the software (domain, program, or situation model) (Vans et al., 1999), which then drives the comprehension process (Burkhardt et al., 2002). Because the domain model is linked to what the software accomplishes, software maintainers tend to invoke it when they are familiar with the application domain (von Mayrhauser and Vans, 1995). When maintainers lack application domain knowledge, they tend to invoke the program model thereby relying on their understanding of the programming language and standard programming constructs (von Mayrhauser and Vans, 1995). Because the situation model tends to develop after the program model, and only after extensive interaction with a piece of software, it is unlikely to be invoked at the outset of comprehension (von Mayrhauser et al., 1997).

Software modification tasks can be conceptualised as emphasising one of the types of information embedded within software. In this study, we investigated two types of tasks that have been examined in prior modification studies: a control flow modification task and a function modification task. A control flow modification task should be consistent with a program model because it emphasises the way in which software accomplishes tasks (von Mayrhauser and Vans, 1995), while a function modification task should be consistent with a domain model because it emphasises function information (Vans et al., 1999).

We conceptualise software maintainers as creating a *mental representation of the IS task* based on the requirements of the modification task, the way in which the modification task is presented (*external IS problem representation*), and their existing knowledge of software development (*internal representation of the application domain*). The external representation of the modification (*external IS problem representation*) is a specification of the software modification task such as a narrative or graphic. Essentially, then, the maintainer's *mental representation of the IS task* will emphasise the type of knowledge that is emphasised in the task requirements (*IS task*). The maintainer's *mental representation of the software (application domain)* and *mental representation of the modification task (IS task)* may or may not match in that they may or may not emphasise the same type of information.

## Operationalisation of the study and study findings

This study operationalised cognitive fit by crossing familiarity with the application domain with the type of modification task. A maintainer's domain model was invoked by using software from a familiar application domain (Vans et al., 1999) and their program model by using software from an unfamiliar application domain (von Mayrhauser and Vans, 1995). A maintainer's *mental*

*representation of the IS task* was invoked using a modification task that emphasised function information or one that emphasised control flow information. Hence cognitive fit exists when maintainers conduct either a function task in a familiar application domain or a control flow task in an unfamiliar application domain. Such a match facilitates *problem-solving performance*. Correspondingly, cognitive fit does not exist when a control flow task is conducted in a familiar application domain and a function task in an unfamiliar application domain. This mismatch results in dual-task interference, which results in lower *problem-solving performance*.

The theory presented suggests that the fit conditions established would moderate the effectiveness of the comprehension that occurs during problem solving, which is reflected in changes to the *mental representation of the software (application domain)*. Hence the dependent variable in the study was change in the level of comprehension observed during the modification task. The data analysis therefore evaluated the three-way relationship between application domain familiarity, type of modification task, and changes in comprehension.

The propositions presented earlier were supported in this study; that is, the three-way interaction between application domain familiarity, type of modification task, and changes in comprehension was significant. As shown in Figure 2, when cognitive fit exists (familiar application domain/function modification and unfamiliar application domain/control flow modification), the relationships between percent change in comprehension and modification performance are positive; that is, increases in comprehension of the software are associated with higher levels of performance on the modification tasks, and Proposition IS-1 is supported. In contrast, as shown in Figure 3, when cognitive fit does not exist (familiar application domain/control flow modification and unfamiliar application domain/function modification) the relationships are negative, demonstrating dual-task interference; that is, increases in comprehension are associated with lower levels of modification performance. Proposition IS-2 is supported.

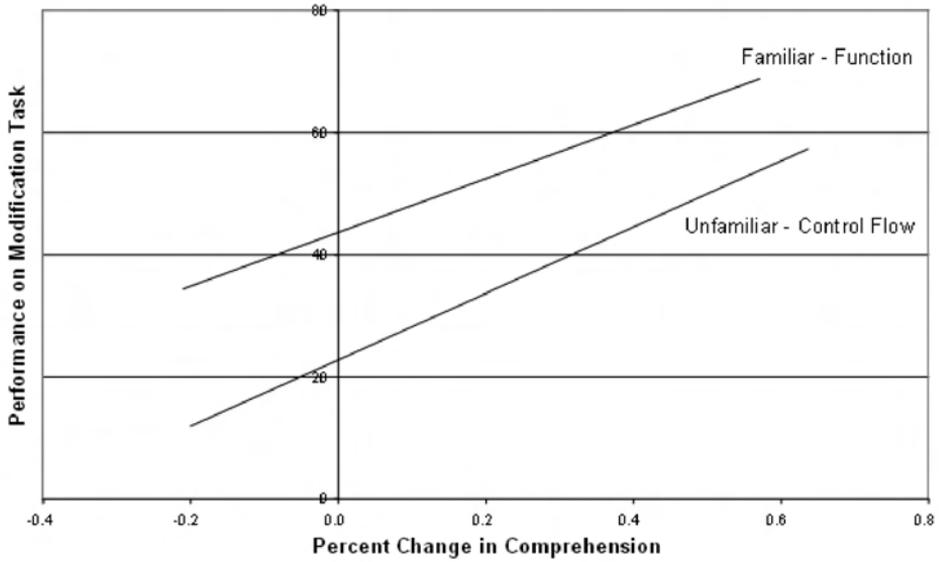


Figure 2: Relationship between percent change in comprehension and performance on modification task in conditions of cognitive fit.

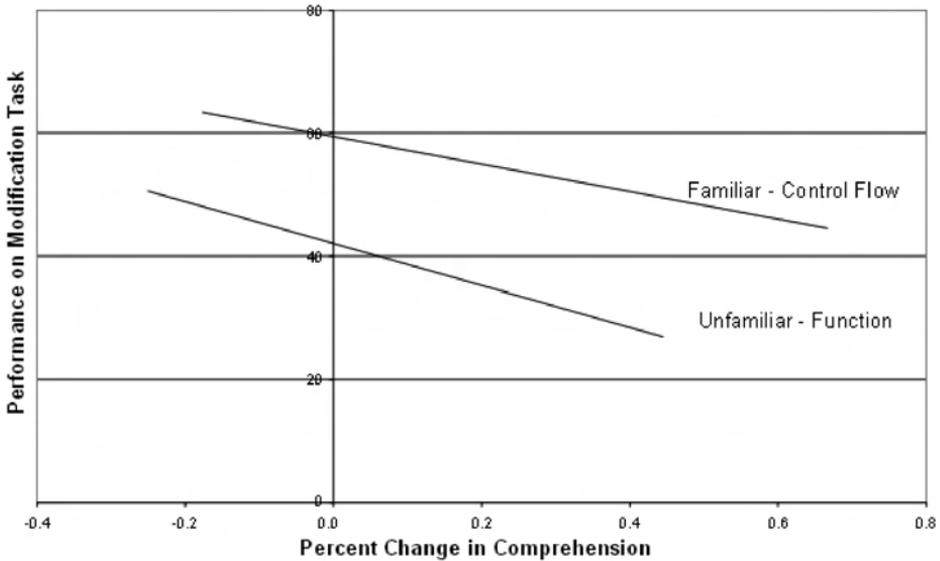


Figure 3: Relationship between percentage change in comprehension and performance on modification task in conditions where cognitive fit does not exist.

## Discussion and implications

Although the application domain has long been acknowledged as playing a significant role in IS problem solving, very little research has been conducted into the effect that it has on performance on IS tasks, and even less theory has been developed. In this paper, we present theory that explains the role of application domain knowledge that is contingent upon the structured nature of the IS task under consideration. We then illustrate the theory on the well-structured problem area of conceptual schema understanding, and on the ill-structured problem area of software maintenance.

## Discussion of the findings

This research develops a unifying theory of the role of the application domain in IS problem solving that explains the findings from two experiments that focused on the role of the application domain in IS problem solving and that produced different results.

The theoretical framework that we use to form the structure for our theory is a dual-task problem-solving model based on the theory of cognitive fit. Cognitive fit applies not only to problem solving in each of the contributing domains (application and IS), but also to the interaction between the two. The theory of cognitive fit allows us to distinguish different types of interactions between the tasks in the IS and application domains, when the two types of tasks match and when they do not. Those interactions may be supportive, neutral, or conflicting, depending on the structured nature of the problem area under investigation.

In solving tasks in well-structured problem areas, all of the information needed for problem solution is available in the *external problem representation* and problem solving can take place with reference to IS domain knowledge alone. In this case, knowledge of the application domain plays a role only in solving problems in which cognitive fit does not exist. Analysis of the well-structured problem-solving area of conceptual schema understanding (Khatri et al., 2006) revealed that knowledge of the application domain aided problem solving only in schema-based problem-solving tasks (fit does not exist), and not in syntactic and semantic comprehension tasks (fit exists). When cognitive fit does not exist, the information required for task solution is not available directly in the conceptual schema and transformations are required.

In solving tasks in ill-structured problem areas, on the other hand, the information needed for problem solution is not available in the *external problem representation* and application domain knowledge is essential to problem solution. When knowledge of the application domain matches the knowledge required to solve the problem, cognitive fit exists and problem solving is facilitated. However, when knowledge of the application domain does not match that required to solve the problem, dual-task interference, which is manifested in an

inverse relationship between knowledge of the software gained during problem solving and performance on the IS task, occurs. Analysis of software maintenance tasks (Shaft and Vessey, 2006) revealed that when knowledge of the application domain matched that required to solve the maintenance task, improved knowledge of the application domain during conduct of the modification task was linked to better problem-solving performance. However, when the two types of knowledge did not match, an inverse relationship between knowledge of the application domain and problem-solving performance resulted.

## Implications and future research directions

Our theory has implications for research in both IS and cognitive psychology. From the viewpoint of research in IS, there are two major implications. First, the dual-task problem-solving model presents a new way of viewing IS problem-solving. Its foundation in theory in cognitive psychology provides the opportunity for IS researchers to investigate the role of what is acknowledged to be an important and under-researched area of IS problem solving: the role of the application domain. The dual-task problem-solving model and its theoretical underpinnings therefore open the way for the development of a stream of research on the role of the application domain. It should always be remembered, however, that the research needs to be conducted in the context of the degree of structure in the problem area under investigation.

Second, this research adds to the strength of a growing body of literature that further testifies to the pervasiveness of cognitive fit in problem solving (see Vessey, 2006). There are a number of possible avenues for further investigation. For example, the distributed model of problem-solving suggests other factors in the fit models, such as the nature of the internal and mental representations in each of the domains, may either facilitate or inhibit problem solving in a given set of circumstances, and could be the subject of future research.

From the viewpoint of research in cognitive psychology, the community has focused on 'people's ability (or inability) to perform two or more activities concurrently' (Pashler, 1994). The findings of dual-task interference have been pervasive, and research does not appear to have been undertaken to examine other possible types of interactions (and their underlying mechanisms), although a number of authors have observed that problem solvers have a greater ability to perform two tasks that are compatible, as opposed to incompatible, at the same time, thus reducing the impact of dual-task interference (see, for example, Koch and Prinz, 2002; Whitaker, 1979).

What is specific to the types of tasks we investigated is that they either interact with each other, or have the potential to interact. Therefore, instead of focusing on the mechanisms by which dual-task interference occurs, we focused on the circumstances in which knowledge in each of the contributing domains interacts,

and the type of interaction that results. Our contribution to the theory of dual-task problem solving in general, therefore, lies in introducing theory to determine when dual-task problem solving results in synergies between the two types of interacting tasks, when it results in interference, and when there are no effects. The cognitive psychology community could extend the focus of its research to determine the mechanisms by which certain tasks that are conducted simultaneously facilitate, while others inhibit, problem solving.

## Conclusion

This research presents theory that formalises, and generalises to tasks in well-structured and ill-structured problem areas the role of the application domain in IS problem solving, thus opening the way to furthering knowledge on this important aspect of IS problem solving. The dual-task problem-solving model that forms the overarching framework for the theory was inspired by research on dual-task interference in cognitive psychology. The true contribution of this model lies in the use of the theory of cognitive fit to determine the different types of interactions that may arise.

## References

- Alexander, P. A. 1992, 'Domain knowledge: evolving themes and emerging concerns', *Educational Psychologist*, vol. 27, no. 1, pp. 33-51.
- Alexander, P. A. and Judy, J. E. 1988, 'The interaction of domain-specific and strategic knowledge in academic performance', *Review of Educational Research*, vol. 58, no. 4, pp. 375-404.
- Blum, B. A. 1989, 'A paradigm for the 1990s validated in the 1980s', *Proceedings of the AIAA Conference*, pp. 502-11.
- Bodart, F., Patel, A., Sim, M. and Weber, R. 2001, 'Should optional properties be used in conceptual modelling? a theory and three empirical tests', *Information Systems Research*, vol. 12, no. 4, pp. 384-405.
- Broadbent, D. E. 1971, *Decision and Stress*, London: Academic Press.
- Brooks, F. P. 1987, 'No silver bullet: essence and accidents of software engineering', *IEEE Computer*, vol. 20, no. 4, pp. 10-9.
- Burkhardt, J., De tienne, F. and Wiedenbeck, S. 2002, 'Object-oriented program comprehension: effect of expertise, task and phase', *Empirical Software Engineering*, vol. 7, no. 2, pp. 115-56.
- Burton-Jones, A. and Weber, R. 1999, 'Understanding relationships with attributes in entity-relationship diagrams', *Proceedings of the Twentieth International Conference on Information Systems*, pp. 214-28.
- Chen, P. P. 1976, 'The Entity-Relationship Model — Toward a unified view of data', *ACM Transactions of Database Systems*, vol. 1, no. 4, pp. 9-36.

- Chi, M. T. H. and Glaser, R. 1985, 'Problem solving ability', in Sternberg R. J. (ed.), *Human Abilities: An Information Processing Approach*, NY: W. H. Freeman and Company.
- Codd, E. F. 1970, 'A relational model of data for large shared data banks', *Communications of the ACM*, vol. 13, no. 6, pp. 377-87.
- Corritore, C. L. and Wiedenbeck, S. 1999, 'Mental representations of expert procedural and object-oriented programmers in a software maintenance task', *International Journal of Human-Computer Studies*, vol. 50, pp. 61-83.
- Curtis, B., Krasner, H. and Iscoe, N. 1988, 'A field study of the software design process for large scale systems', *Communications of the ACM*, vol. 31, no. 11, pp. 1268-87.
- Durso, F. T., Hackworth, C., Truitt, T. R., Crutchfield, J., Nikolic, D. and Manning, C. A. 1998, 'Situation awareness as a predictor of performance in en route air traffic controllers', *Air Traffic Control Quarterly*, vol. 6, no. 1.
- Durso, F. T. and Gronlund, S. D. 1999, 'Situation awareness', in Durso, F. T., Nickerson, R. S., Schvaneveldt, R. W., Dumais, S. T., Lindsay, D. S. and Chi, M. T. (eds), *Handbook of Applied Cognition*, John Wiley and Sons.
- Elmasri, R. and Navathe, S. B. 1994, *Fundamentals of Database Systems*, (2nd ed.) Benjamin/Cummings Publishing Co., Redwood City, CA.
- Gemino, A. 1999, 'Empirical methods for comparing system analysis modelling techniques', Unpublished PhD thesis, University of British Columbia, Vancouver, B.C., Canada.
- Gemino, A. and Wand, Y. 2003, 'Evaluating modelling techniques based on models of learning', *Communications of the ACM*, vol. 46, no. 10, pp. 79-84.
- Glass, R. L. and Vessey, I. 1992, 'Toward a taxonomy of application domains: history', *Journal of Systems and Software*, vol. 17, no. 2, pp. 189-99.
- Greeno, J. 1978, 'Natures of problem-solving abilities', in Estes W. (ed.) *Handbook of Learning and Cognitive Processes*, vol. 5, Hillsdale, NJ: Lawrence Erlbaum Associates.
- Hogg, D. N., Folleso, K., Strand-Volden, F. and Torralba, B. 1995, 'Development of a situation awareness measure to evaluate advanced alarm systems in nuclear power plant control rooms', *Ergonomics*, vol. 11, pp. 394-413.
- Kahneman, D. 1973, *Attention and Effort*, Englewood Cliffs, NJ: Prentice-Hall.

- Khatri, V., Vessey, I., Ramesh, V., Clay, P. and Park, S-J. 2006, 'Comprehension of conceptual schemas: Exploring the role of application and IS domain knowledge', *Information Systems Research*, vol. 17, no. 1, pp. 81-99.
- Kitchner, K. S., 1983, 'Cognition, metacognition, and epistemistic cognition: A three-level model of cognitive processing', *Human Development*, vol. 26, pp. 222-32.
- Koch, I. and Prinz, W. 2002, 'Process interference and code overlap in dual-task performance', *Journal of Experimental Psychology: Human Perception and Performance*, vol. 28, no. 1, pp. 192-201.
- Mayer, R. E. 1991, *Thinking, Problem Solving, Cognition*, W. H. Freeman and Company, New York, NY.
- McPeck, H. 1990, 'Critical thinking and subject specificity: a reply to Ennis', *Educational Researcher*, vol. 19, no. 4, pp. 10-2.
- Navon, D. 1990, 'Exploring two methods for estimating performance tradeoff', *Bulletin of the Psychonomic Society*, vol. 28, no. 2, pp. 155-7.
- Navon, D. and Gopher, D. 1979, 'On the economy of the human processing systems', *Psychological Review*, vol. 86, pp. 254-5.
- Navon, D. and Miller, J. 1987, 'Role of outcome conflict in dual-task interference', *Journal of Experimental Psychology: Human Perception and Performance*, vol. 13, no. 3, pp. 435-48.
- Pashler, H. 1994, 'Dual-task interference in simple tasks: Data and theory', *Psychological Bulletin*, vol. 116, no. 2, pp. 220-44.
- Pashler, H. and O'Brien, S. 1993, 'Dual-task interference and the cerebral hemispheres', *Journal of Experimental Psychology-Human Perception and Performance*, vol. 19, no. 2, pp. 315-30.
- Pennington, N. 1987a, 'Stimulus structures and mental representations in expert comprehension of computer programs', *Cognitive Psychology*, vol. 19, pp. 295-341.
- Pennington, N. 1987b, 'Comprehension strategies in programming', in Olson G. M., Sheppard S., and Soloway, E. (eds), *Empirical Studies of Programmers: First Workshop*, Ablex Publishing, Norwood, NJ.
- Purao, S., Rossi, M. and Bush, A. 2002, 'Toward an understanding of the use of problem and design spaces during object-oriented system development', *Information and Organisation*, vol. 12, pp. 249-81.
- Reitman, W. R. 1964, 'Heuristic decision procedures, open constraints, and the structure of ill-defined problems', in Shelly, M. W. and Bryan, G. L. (eds), *Human Judgements and Optimality*, New York: John Wiley and Sons.

- Shaft, T. M. and Vessey, I. 1995, 'The relevance of application domain knowledge: The case of computer program comprehension', *Information Systems Research*, vol. 6, no. 3, pp. 286-99.
- Shaft, T. M. and Vessey, I. 1998, 'The relevance of application domain knowledge: Characterising the computer program comprehension process', *The Journal of Management Information Systems*, vol. 15, no. 1, pp. 51-78.
- Shaft, T. M. and Vessey, I. 2006, 'The role of cognitive fit in the relationship between software comprehension and modification', *MIS Quarterly*, vol. 30, no. 1, pp. 29-55.
- Shanks, G., Tansley, E., Nuredini, J., Tobin, D, Moody, D. and Weber, R. 2002, 'Representing part-whole relationships in conceptual modelling: An empirical evaluation', *Proceedings of the Twenty-Third International Conference on Information Systems*, pp. 89-100.
- Shanks, G., Nuredini, J., Tobin, D, Moody, D. and Weber, R. 2003, 'Representing things and properties in conceptual modelling: An empirical investigation', *European Conference on Information Systems*.
- Shneiderman, B. and Mayer, R. E. 1979, 'Syntactic/semantic interactions in programmer behavior: A model and experimental results', *International Journal of Computer and Information Science*, vol. 8, pp. 219-38.
- Sinnott, J. D., 1989, 'A model for solution of ill-structured problems: Implications for everyday and abstract problem solving', in Sinnott, J. D. (ed.), *Everyday problem solving: Theory and application*, Praeger, New York.
- Taylor, R. M., Finnie, S. and Hoy, C. 1997, 'Cognitive rigidity: The effects of mission planning and automation on cognitive control in dynamic situations', 9<sup>th</sup> International Symposium on Aviation Psychology, Columbus, OH.
- Vans, A. M., von Mayrhauser, A. and Somlo, G. 1999, 'Program understanding behavior during corrective maintenance of large-scale software', *International Journal of Human-Computer Studies*, vol. 51, pp. 31-70.
- Van Selst, M. and Jolicoeur, P. 1997, 'Decision and response in dual-task interference', *Cognitive Psychology*, vol. 33, no. 3, pp. 266-307.
- Vessey, I. 1991, 'Cognitive fit: A theory-based analysis of the graph versus tables literature', *Decision Sciences*, vol. 22, no. 2, pp. 219-40.
- Vessey, I. 1994, 'The effect of information presentation on decision making: An analysis using cost-benefit theory', *Information and Management*, vol. 27, pp. 103-19.
- Vessey, I. 2006, 'The theory of cognitive fit: One aspect of a general theory of problem solving?', in Zhang, P. and Galletta, D. (eds), *Human-computer*

*interaction and management information systems: Foundations*, Advances in Management Information Systems Series, Armonk, NY: M.E. Sharpe.

- Vessey, I. and Conger, S. 1993, 'Learning to specify information requirements: The relationship between application and methodology', *Journal of Management Information Systems*, vol. 10, no. 2, pp. 177-201.
- von Mayrhauser, A. and Vans, A. M. 1995, 'Industrial experience with an integrated code comprehension model', *Software Engineering Journal*, vol. 22, no. 6, pp. 171-82.
- Von Mayrhauser, A. and Vans, A. M. 1996, 'Identification of dynamic comprehension processes during large scale maintenance', *IEEE Transactions on Software Engineering*, vol. 22, no. 6, pp. 424-37.
- von Mayrhauser, A., Vans, A. M. and Howe, A. E. 1997, 'Program understanding behavior during enhancement of large-scale software', *Software Maintenance: Research and Practice*, vol. 9, pp. 299-327.
- Voss, J. F. and Post, T. A., 1988, 'On the solving of ill-structured problems', in Chi M. H., Glaser, R. and Farr M. J. (eds), *The Nature of Expertise*, Hillsdale, NJ: Lawrence Erlbaum Associates.
- Whitaker, L. A. 1979, 'Dual-task interference as a function of cognitive processing load', *Acta Psychologica*, vol. 43, no. 1, pp. 71-84.
- Wickens, C. D. 2002, 'Multiple resources and performance prediction', *Theoretical Issues in Ergonomic Science*, vol. 3, no. 2, pp. 159-77.
- Zhang, J. 1997, 'The nature of external representations in problem solving', *Cognitive Science*, vol. 21, no. 2, pp. 179-217.
- Zhang, J. and Norman, D. A. 1994, 'Representations in distributed cognitive tasks', *Cognitive Science*, vol. 57, pp. 87-122.