KNOWLEDGE OVERLAP, TASK INTERDEPENDENCE, AND TRUST IN IS DEVELOPMENT

Jungpil Hahn

Assistant Professor of Management
Krannert School of Management
Purdue University
jphahn@purdue.edu

and

Gwanhoo Lee

Associate Professor of Information Technology
Director, Center for IT and the Global Economy
Kogod School of Business
American University
glee@american.edu

Last Revised: October, 2010

ABSTRACT

Information systems development (ISD) is fundamentally a search process by which the team seeks to find an optimal system configuration that produces the highest performance. As information systems are embodiments of business-domain knowledge and technical knowledge, ISD requires both. The business unit is ultimately responsible for making business design choices whereas the IS unit is largely responsible for making technical design choices. Complexity in ISD arises when these design choices are interdependent. We argue that knowledge overlaps between business and IS play an important role in the ISD process. Using an NK fitness landscapes model of ISD, this research investigates how knowledge overlaps influence ISD performance (1) when the level of interdependencies among design choices varies, (2) for different distributions of within-unit and between-unit interdependencies, (3) when between-unit interdependencies are balanced or skewed, and (4) when inter-unit trust exists or is absent. We report the results of a simulation study and discuss their implications and insights.

Keywords: information systems development (ISD), complexity, knowledge overlap, trust, complex adaptive systems, *NK* fitness landscape model, simulation

INTRODUCTION

Information systems development (ISD) typically involves not only technical specialists but also end-users from business units (Tiwana 2009). ISD requires both technical knowledge and business domain knowledge to search for an optimal configuration of the system that best meets user requirements with the best possible technological architecture and capabilities. An information system is an embodiment of business-domain expertise and technical expertise (Tiwana 2004).

A system configuration comprises of a series of design choices. Some choices determine how business process and business model are configured and embodied in the system, while others determine how hardware, software, and network technologies are configured and implemented. Conventionally, business configuration choices are made during the user requirement determination phase and technology configuration choices are made during the system design and development phases. With iterative and agile development methods (Beck and Andres 2005; Lee and Xia 2010), these choices are made through a number of incremental iterations between end-users and IT professionals.

ISD is complex because business configuration choices and technology configuration choices are often interdependent. For example, the business value of a business configuration choice may depend on other business configuration choices as well as other technology configuration choices. Similarly, the business value of a technology configuration choice may depend on other technology choices and business choices. Therefore, ISD can be viewed as a search process by which the team utilizes business domain knowledge and technical knowledge to find an optimal set of business configuration choices and technology configuration choices that are interdependent.

Naturally, knowledge plays an important role in ISD. The need for combining and integrating business knowledge and technical knowledge complicates the ISD process because knowledge is not evenly distributed across the organization (Tiwana 2004). Business domain knowledge is mostly distributed to the business unit whereas technical knowledge is mostly distributed to the IS unit. The IS unit generally lacks business domain knowledge, and the business unit lacks technical knowledge. Such knowledge asymmetry can be problematic because it creates knowledge barriers that inhibit the adoption of complex technologies (Attewell 1992). Furthermore, it is not always easy to transfer knowledge between the IS unit and the business unit (Ko et al. 2005).

However, the division of knowledge to specialized units is not always so clear-cut. It has become increasingly common to find business units to possess ample technological knowledge and IS units to have knowledge of business domains, circumstances we refer to as *knowledge overlaps*. Prior research suggests that knowledge overlap between the IS unit and the business unit is desirable and is associated with higher performance of the ISD process (Espinosa et al. 2007; Tiwana, 2009). In the ISD process, a group of project team members are responsible for determining user requirements whereas another group of project team members are responsible for technical design and implementation. Task partitioning arises because the two groups tend to possess different sets of knowledge and competencies (von Hippel 1990) – the former group consists mainly of users and business analysts who have business domain knowledge, whereas the latter group consists mainly of system analysts and developers who possess technical expertise. While the notion of knowledge overlap has been widely acknowledged in the literature, its effects are not yet well understood. For example, how much knowledge overlap is necessary (or sufficient) for effective ISD? When is knowledge overlap more beneficial to ISD performance? Does knowledge overlap need to be evenly distributed across the business and IS units? These are some of the important questions for which the current literature does not provide clear answers.

The purpose of this study is to theoretically explore how knowledge overlap between the two groups in the project team influences ISD performance. More specifically, the objectives of this research are four-fold; it investigates how differently knowledge overlap affects the business value of the system (1) when the level of interdependencies among design choices varies, (2) for different distributions of within-unit and between-unit interdependencies, (3) when between-unit interdependencies are balanced or skewed, and (4) when inter-unit trust exists or is absent.

We use the complex adaptive systems approach, more specifically the *NK* fitness landscapes model, to address the research questions outlined above. One of the advantages of the *NK* fitness landscape simulation modeling method is that it allows researchers to examine complex theoretical relationships that analytical models and/or field studies cannot fully uncover (Miller and Page 2007). Analytical models rely on drastically simplified representations of organizations for analytical tractability and as a result cannot faithfully represent the richness of actual organizations. Field studies of actual organizations enjoy the luxury of realism, but it is oftentimes difficult, if not impossible, to observe, measure or manipulate the theoretical constructs of interest. The complex adaptive systems approach follows a smaller but long tradition of computational models of organizations and organizational processes (e.g., Cyert and March 1963; Cohen et al. 1972; March 1991). The agent-based simulation modeling methodology enables us to incorporate a greater number of interdependent elements in a formal model than is possible with a closed-form analytical approach (Davis et al. 2007). In addition, the simulation methodology allows us to freely manipulate the theoretical constructs of interest to help acquire non-intuitive and/or nuanced theoretical insights through various combinations of experimental conditions, which is not possible with field studies.

In the following sections, we discuss the prior literature relevant to our research. We then explain our *NK* landscapes model of information systems development (ISD) and experimental design. We discuss the results and their implications and conclude by proposing a set of propositions based on our findings and observations.

2. THEORETICAL BACKGROUND

2.1. Knowledge Overlap

Prior research has argued that knowledge overlap, or shared knowledge between the business unit and the IS unit, improves various dimensions of IS performance (Tiwana 2009; Nelson and Cooprider 1996; Bassellier and Benbasat 2004; Bassellier et al. 2003; Espinosa et al. 2007; Reich and Benbasat 2000). For example, shared knowledge was found to influence business-IT alignment (Reich and

Benbasat 2000), business-IT partnerships (Nelson and Cooprider 1996), ISD coordination (Espinosa et al. 2007), and ISD project decision control and management (Tiwana 2009). Knowledge overlap is important because individuals need to have a certain level of overlap in their individual knowledge bases to effectively coordinate collective action (Alavi and Leidner 2001). Shared knowledge develops over time from prior familiarity with common tools and processes, the system being developed, the task domain, and team members (Espinosa et al. 2007).

The business knowledge of IT professionals enables them to participate in important organizational decision-making processes (Feeny and Willcocks 1998). Important business knowledge for IT professionals includes the knowledge about organizational overview, organizational units, organizational responsibility, and IT–business integration (Bassellier and Benbasat 2004). In particular, the knowledge of IT-business integration enables IT professionals to identify synergies between IT and business activities and to understand how various business and technical parts fit together (Bassellier and Benbasat 2004).

Similarly, the technical knowledge of business managers and users is also important for successful ISD. Increasingly, business managers are expected to assume ownership of ISD projects and to take a leadership role for them (Rockart et al. 1996). Important IT knowledge that business managers need to acquire includes the knowledge of technology, applications, systems development, and management of IT (Bassellier and Benbasat 2001). Prior research shows that IT knowledge is positively associated with business managers' intentions to champion IT (Bassellier et al., 2003).

While the literature seems to suggest that knowledge overlap is generally beneficial and desirable, an important gap in prior literature is that little is known about the conditions under which knowledge overlap is more or less beneficial to ISD performance. One exception is Tiwana (2004)'s study on the moderating effect of project and process novelty on the relationship between knowledge overlap and outsourcing ISD performance. He found that effective outsourcing requires a good fit in terms of the business and technical knowledge with project and process novelty across the client-vendor dyad and cautioned that blindly pursuing knowledge overlaps can be perilous to ISD performance.

2.2. Task Interdependence

The theoretical underpinning for task interdependence can be found in the task complexity literature. The complexity of a task such as ISD increases when multiple components exist, when these components are interdependent, and when they change over time (Xia and Lee 2005; Wood 1986; Ribbers and Schoo 2002). The problems in building complex systems often arise in the interfaces between hardware, software, and human components (Leveson 1997). Interdependencies among these elements make it difficult to predict the project's process and outcome (Wood 1986). One important gap in the task complexity literature is that the effect of different patterns of interdependencies on ISD performance has not been understood. In the ISD process, business requirements and technological choices often interact with one another and these interdependencies may exhibit different patterns depending on types of information systems and business application problems. For example, some systems may be largely modular in the sense that while interdependencies may exist within business or technology domains, interdependencies may not exist (or may trivially exist) across domains. Alternatively, some systems may exhibit greater interdependencies across domains than within.

2.3. Trust and Knowledge Transfer

Prior literature suggests that trust may play an important role in knowledge transfer between two units. Trust reduces complexity by enabling units with different knowledge bases to collaborate (Gefen 2000). For example, it has been found that trust is positively associated with virtual collaborative relationship performance (Paul and McDanniel 2004). In addition, trust may facilitate the learning and innovation (Sako 1998). Competence trust among other types of trust is most relevant to the current research. Competence trust is a belief about whether the other unit is capable of doing what it says it will do (Sako 1998; Mayer et al. 1995). It is an assessment of the expertise and abilities of the other units.

Competence trust is required in complexity reducing collaborative efforts when the required skills are not found within one unit (Newell and Swan 2000).

In order for knowledge overlap to effectively influence the ISD process, overlapping knowledge and decisions based on such knowledge need to be transferred from one unit to the other unit. It has been argued that knowledge transfer occurs when a contributor shares knowledge that is *used* by an adopter (Darr and Kurtzberg 2000). Depending on the level of trust, one unit may or may not be willing to accept the other unit's decision despite the other unit's input arising from overlapping knowledge. Despite the apparent relevance of trust in ISD, especially with respect to knowledge overlaps, an important gap is that no prior research has examined how trust may moderate the effect of knowledge overlap on ISD performance.

3. MODEL

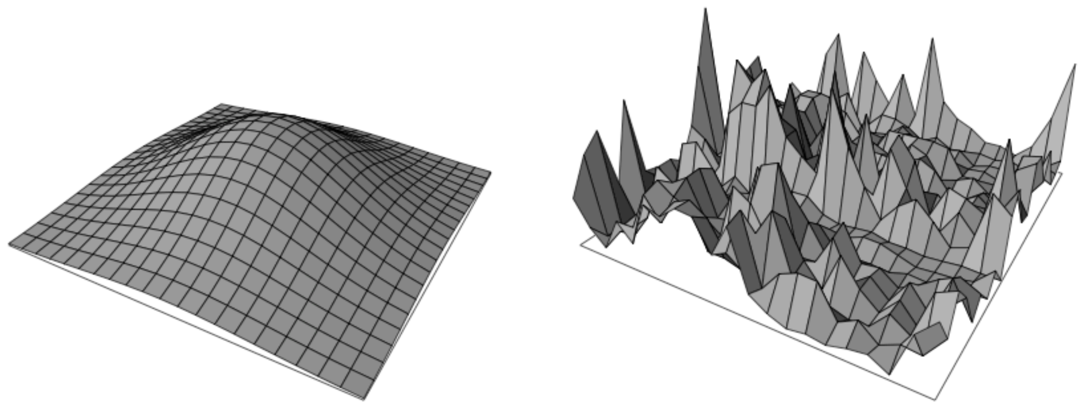
Our model is an extension of the *NK* fitness landscape model. The *NK* model developed by Kauffman (1993) provides a simple, yet powerful analytical framework to study complex adaptive systems. Although the model itself was developed for the study of evolutionary biology, it has extensively been applied to management research to study organizational adaptation (Levinthal 1997), the impact of modularity on innovation and imitation (Ethiraj and Levinthal 2003; Ethiraj, Levinthal and Roy 2008), the efficacy of different organizational search strategies (Gavetti and Levinthal 2000), the efficacy of different organizational designs (Rivkin and Siggelkow 2003; Siggelkow and Rivkin 2005), open vs. closed innovation (Almirall and Casadesus-Masanell 2010), etc. Although, the *NK* modeling framework has recently gained broad acceptance within the strategic management and organization sciences literatures, it is still relatively unknown within the IS literature. Accordingly, we briefly review the fundamental concepts of the *NK* fitness landscapes model before proceeding with details of our model.[1]

[1] The interested reader is directed to Davis et al. (2007, pp. 487-488) for a brief overview of the *NK* modeling approach. More technical details of the approach can be found in Kauffman (1989; 1993).

3.1. The NK Fitness Landscapes Model

In the *NK* model, a complex adaptive system (e.g., strategies, products, projects, etc.) is conceptualized as involving *N* decision variables and *K* interactions among these decision variables. Each configuration of a set of decision variables is associated with a fitness value, which can be interpreted as performance if that particular configuration is implemented. The system uses search strategies (e.g., incremental/local hill-climbing, trial and error search, long jumps, etc.) to navigate within the fitness landscape to find positions of greatest fitness (i.e., best strategy, best product, best project, etc.). These search strategies are heuristics/routines the system uses to configure (and reconfigure) the values of the *N* decision variables. The two parameters (*N* and *K*) of the *NK* model allow the modeler to create "tunable" fitness landscapes of the decision environment of varying degrees of complexity on which to test the efficacy of various search strategies. When there is little interaction among decision variables (i.e., low *K*), the resulting fitness landscape is "smooth" (see Figure 1a). Conversely, when the decision variables become highly interdependent (i.e., high *K*), the resulting fitness landscape is "rugged" and multipeaked (see Figure 1b). The complexity (or ruggedness) of the fitness landscape determines the efficacy of the various search strategies. For example, in an environment of low complexity (i.e., within a smooth fitness landscape), incremental hill-climbing strategies work very well and the system will eventually find the position with globally optimal fitness level. However, in a complex environment (i.e., within a rugged fitness landscape), search for a high position is profoundly more difficult as incremental local search strategies are prone to lead the system to get stuck in basins of attraction – a local peak/optimum, but not necessarily the global optimum (Kauffman 1989).

While early management research that used the *NK* modeling framework focused on analyzing and comparing the efficacy of various search strategies, recent efforts have shifted the focus of attention to issues related to organizational design. For example, Siggelkow and Rivkin (2005) analyze the effects of departmental information processing power, incentive structures, cross-departmental coordination and information flow on the speed and diversity of organizational search. The model of information systems development, which we will present next also extends this line of research by investigating the impact of various degrees and patterns of knowledge overlap between the business unit and the IS unit, the extent and pattern of interdependencies between the decisions of the business unit and those of the IS unit, and the presence of trust leading to effective knowledge transfer between the business units.



(a) "Smooth" Fitness Landscape (b) "Rugged" Fitness Landscape

Figure 1. Fitness Landscapes

3.2. A Model of Information Systems Development

To capture the essence of the information systems development (ISD) process in its realistic yet parsimonious form, we model an ISD project as consisting of *N* design choices (features or decision variables), where each design choice can take one of two values – 0 or 1.[2] A design choice could represent a business domain related decision (e.g., make or buy intermediate products) or a technology related decision (e.g., distribute or centralize the database). Design choices can interact in the sense that the value contribution of one decision may depend on the configuration of some other decision. Such interdependencies can exist within the business or IS domains (e.g., the <u>business</u> decision to make or buy may depend on a related <u>business</u> decision to acquire or develop complementary skills) or across domains

[2] The *NK* model has been shown to be robust to this simplification (Kauffman 1989). The model can be extended to an arbitrary finite number of possible values of an attribute without altering the qualitative properties of the model.

(e.g., the technical decision to centralize or decentralize the enterprise data may depend on a related business decision to centralize management authority at the headquarters or empower regional branch managers).

With this setup, the ISD project can be represented as an N element vector of decisions: $\mathbf{d} = \langle d_1, d_2, \ldots, d_N \rangle$, which as a result can take on 2^N possible configurations. Each ISD project configuration is associated with a fitness value, which can be interpreted as performance if that particular configuration of ISD project is implemented. Two ISD projects that arrive at the same configuration of design choices are presumed to achieve the same level of performance. The objective of the ISD process is to find the highest value point in the fitness landscape of ISD project configurations.

3.2.1. Organizational structure: allocation of decisions

As discussed previously, in the ISD project some design choices determine how business processes and business models are configured and embodied in the system, whereas others determine how hardware, software, and network technologies are configured and implemented. In ISD projects, these design choices are made by different units due to specialized knowledge (von Hippel 1990) – the business unit is responsible for making decisions about the business domain, whereas the IS unit is responsible for those about the technology domain.

To incorporate the allocation of decisions to different organizational units, we model the organization as composed of two units – a business unit and an IS unit. Since the ISD project is characterized as consisting of N design choices, each unit (business or IS) is responsible for $N/2$ design choices.[3] Formally, the ISD project configuration vector $\mathbf{d}$ can be partitioned into two subsets $\langle \mathbf{d_{bus}}, \mathbf{d_{IS}} \rangle$.

[3] For simplicity, we assume that the decisions are equally distributed among the two units – $N/2$ design choices for each of the two units. In reality, some projects may involve a greater number of design choices for either of the two units. Analyzing the impact of the distribution of decisions is beyond the scope of this paper. We leave it to future research to investigate this issue.

Assuming that N is even, the business and IS units are modeled as: $\mathbf{d}_{\mathbf{bus}} = \langle d_1, d_2, \ldots, d_{N/2} \rangle$ and $\mathbf{d}_{\mathbf{IS}} = \langle d_{N/2+1}, d_{N/2+2}, \ldots, d_N \rangle$.[4]

Decision interdependence and ISD project complexity

The interdependencies among design choices may exist within domain and/or across domains. Interdependence exists within domain if the business value of a business (technology) design choice depends on other business (technology) design choices. Conversely, interdependence exists across domains if the business value of a business (technology) design choice depends on other technology (business) design choices.

The efficacy of each decision d_i is affected not only by the choice (0 or 1) for d_i itself but also by the choices of other d_j's that interact with d_i. An $N \times N$ "influence matrix," **INF**, records the interdependencies among decisions (see Figure 2 for some examples of influence matrices for $N = 10$). The i,j^{th} entry of **INF** is marked with an "x" if column decision j influences the contribution of row decision i, or is otherwise left blank.

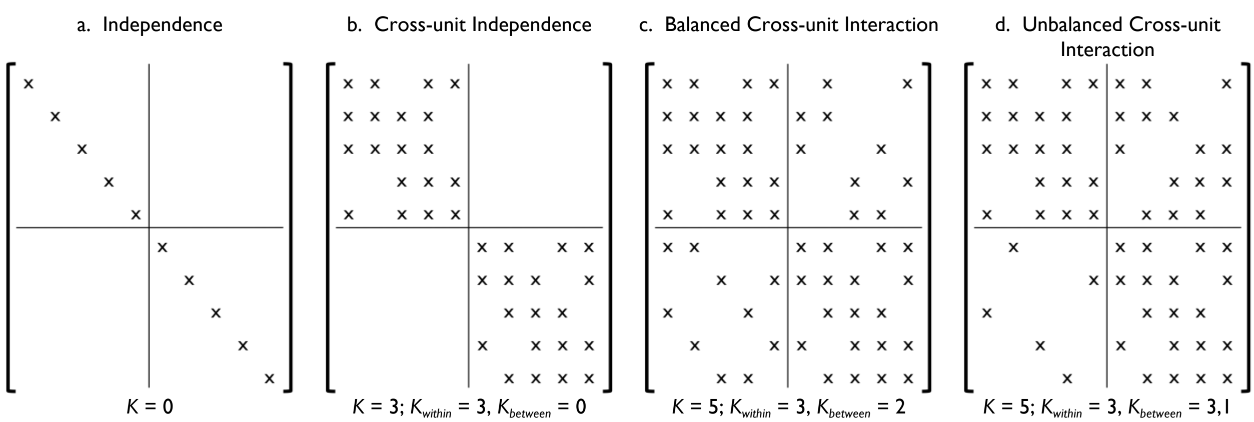


Figure 2. Examples of Influence Matrices (N = 10)

[4] This configuration of decision responsibilities is the baseline case where knowledge overlap does not exist. Later we will extend the model to incorporate knowledge overlaps between the business and IS units.

Given an influence matrix, **INF**, we may stochastically assign a fitness level for each of the 2^N possible configuration of ISD projects. Since the contribution c_i of each decision d_i to the fitness level of the overall ISD project depends on d_j other decisions (i.e., $c_i = c_i(d_i$; other d_j's)), for each possible realization of d_i and the relevant other d_j's, a fitness contribution value is drawn randomly from a uniform distribution on [0, 1] and the overall payoff associated with the configuration of the ISD project <*bus*, *IS*> is the average over N contributions:[5]

$$F\left(< bus, IS >\right) = \sum_{i=1}^{N} c_i \left(d_i ; \text{other } d_j\text{'s}\right) \Big/ N$$

The patterns of choice interdependencies influence the complexity of the ISD project not only by determining the ruggedness of the ISD project fitness landscape (via parameter K) but also by impacting the accuracy of fitness estimates given the allocation of decisions between units and distribution of specialized knowledge across units (Gavetti and Levinthal 2000; Rivkin and Siggelkow 2007; Almirall and Casadesus-Masanell 2010).

3.2.2. Specialized knowledge

The two units involved in the ISD project have specialized knowledge. First, let's consider the case of fully specialized knowledge (i.e., the business unit does not know anything in the technology domain, and the IS unit does not know anything about the business domain); later we will relax this assumption to allow knowledge overlap across units.

Although the ISD project involves setting the configurations for N design choices, since each unit only has knowledge of $N/2$ of the design choice elements, each unit can only infer the performance implications of the design choice elements for which they have knowledge. In other words, although each unit understands the performance implications of their local design choices (i.e., which configuration of business design choices have the highest fitness value given some unknown, not fully understood,

[5] This procedure for generating fitness values given well-controlled patterns of interdependencies is adapted from

configuration of the technology design choices, and vice versa), its understanding of the full implications of their design choices will be imperfect. For example, consider the influence matrix in Figure 2c (balanced cross-unit interaction). If we assume that the business unit is responsible for design choices d_1 through d_5 and the IS unit is responsible for design choices d_6 through d_{10} (where $N = 10$); then although the fitness contribution of the first design choice d_1 depends on d_2, d_4, d_5, d_7 and d_{10}, the business unit only understands the interaction of d_1 with d_2, d_4 and d_5.[6]

We model this imperfect understanding due to knowledge specialization via cognitive simplification (Gavetti and Levinthal 2000). The local assessment of the fitness contribution of a design choice will be the average of the fitness contributions of the known elements across the configurations of the unknown elements. To illustrate, let's reconsider the influence matrix in Figure 2c and suppose the ISD project configuration is specified by the N-dimensional array $\mathbf{d} = (1, 0, 0, 1, 1, 1, 0, 1, 0, 0)$. The fitness contribution of the value of the first element ($d_1 = 1$) of this array depends on the values of the second ($d_2 = 0$), fourth ($d_4 = 1$), fifth ($d_5 = 1$), seventh ($d_7 = 0$) and tenth ($d_{10} = 0$) elements. So even if the fitness contribution of d_1 (given d_2, d_4, and d_5) depends the current (unknown) values of d_7 and d_{10}, the business unit's assessment of c_1 will be the average of the fitness contributions given the four possible combinations of d_7 and d_{10} values (i.e., $<d_7, d_{10}> = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$). See Table 1 for a numerical example.

Table 1. Fitness Contributions Given Knowledge Specialization

ISD Project Configuration	Fitness Contribution of d_1
(1, 0, , 1, 1, , *0*, , , *0*)	0.894
(1, 0, , 1, 1, , *0*, , , *1*)	0.954
(1, 0, , 1, 1, , *1*, , , *0*)	0.127
(1, 0, , 1, 1, , *1*, , , *1*)	0.723
(1, 0, , 1, 1, , ?, , , ?)	avg = 0.674

Rivkin and Siggelkow (2007).

[6] Although the business unit may know the actual value that d_7 and d_{10} take, it does not *understand* how these values interact with the design choice elements within its domain of responsibilities (i.e., d_1, d_2, d_4 and d_5). In other words, in this paper, *knowledge* is equated with *understanding* rather than mere recognition.

3.2.3. Knowledge overlap

If knowledge overlap exists, then the different units understand how some portion of the design choices of the other unit impacts their own decisions. For example, the business unit may understand the fitness implication of a particular technology choice (and vice versa). We model knowledge overlap with parameter (small) $k = \{1,\ldots, N/2\}$, which represents the number of elements in the other unit's knowledge domain, for which the focal unit has knowledge. At the extremes, if $k = 0$, then there is no overlap; if $k = N/2$, then there is full overlap. For each unit, we randomly select k design choice elements from the other unit's set of design choice elements.

To illustrate, let's suppose again that the ISD project configuration is specified by the array $\mathbf{d} = (1, 0, 0, 1, 1, 1, 0, 1, 0, 0)$, that $k = 1$ and that it was (randomly) determined that the business unit also knows d_7. In this case, since the fitness contribution of the value of the first element ($d_1 = 1$) of this array depends on the values of the second ($d_2 = 0$), fourth ($d_4 = 1$), fifth ($d_5 = 1$), seventh ($d_7 = 0$) and tenth ($d_{10} = 0$) elements, and the business unit now understands the fitness contribution of d_1 (given d_2, d_4, d_5, and d_7), the only unknown element will be d_{10}. As a result, the business unit's assessment of c_1 will be the average of the fitness contributions given the two possibilities for d_{10} (i.e., $d_{10} = \{0, 1\}$). See Table 2 for a numerical example.

Table 2. Fitness Contributions Given Knowledge Overlap

ISD Project Configuration	Fitness Contribution of d_1
(1, 0, , 1, 1, , 0, , , *0*)	0.894
(1, 0, , 1, 1, , 0, , , *1*)	0.954
(1, 0, , 1, 1, , 0, , , ?)	avg = 0.924

3.2.4. ISD process as landscape search

The objective of the organization is to find the highest value point in the fitness landscape of ISD project configurations. The organization searches for a good configuration of design choices via an incremental and local experiential search process. This subsection outlines the search process.

Although there are many different approaches to ISD (e.g., sequential waterfall, rapid application development, agile development etc.), in its simplest and generic form, the ISD process can be described as an *incremental* and *iterative* process of requirements determination and systems design. During the requirements determination phase, business design choices are made, while technology design choices are made during the systems design phase. These phases are repeated until a satisfactory design configuration is reached.

Given the complexity of ISD, search is assumed to be local (March and Simon, 1958; Cyert and March, 1963). In other words, organizational units can only effectively consider incremental changes to its current configuration – the units cannot possibly know the performance implications of a random and radically different configuration of design choices (Levinthal 1997). The organizational units are assumed to be able to identify alternative configurations in their immediate neighborhood whose fitness value is superior to their current level of fitness. For example, if the organization's current ISD project configuration is specified by the array $\mathbf{d} = (1, 0, 0, 1, 1, 1, 0, 1, 0, 0)$, with $\mathbf{d_{bus}} = (1, 0, 0, 1, 1)$ and $\mathbf{d_{IS}} = (1, 0, 1, 0, 0)$, then during the requirements determination phase, the *business* unit may consider 5 (i.e., $N/2$) neighboring locations – {(0, 0, 0, 1, 1), (1, 1, 0, 1, 1), (1, 0, 1, 1, 1), (1, 0, 0, 0, 1), (1, 0, 0, 1, 0)} (i.e., permutations of each of the first 5 design choices), while during the systems design phase, the *IS* unit may consider 5 neighboring locations – {(0, 0, 1, 0, 0), (1, 1, 1, 0, 0), (1, 0, 0, 0, 0), (1, 0, 1, 1, 0), (1, 0, 1, 0, 1)} (i.e., permutations of each of the last 5 design choices).[7] While conducting search, the units are not assumed to evaluate all the neighboring alternatives to find the highest fitness level. Rather, the units will *satisfice* by adopting any alternative configuration so long as the fitness value is higher than that of its current configuration. Once a unit has moved to a new location, we switch the focal search unit. In other words, if the business unit has found a new (higher performing) business configuration during the

[7] In the presence of knowledge overlap, each unit will consider $N/2 + k$ neighboring configurations given its knowledge of some (k) portion of the other unit's knowledge. For example, if $k = 1$ and it was randomly determined that the business unit also knows d_7, then the business unit's knowledge set becomes $\mathbf{d_{bus}} = \{d_1, d_2, d_3, d_4, d_5, d_7\}$, and

requirements determination phase, then we move on to the systems design phase where the IS unit can now search for a new (higher performing) technology configuration. If the IS unit finds a new (higher performing) technology configuration during systems design, then the business unit takes over and we repeat the requirements determination phase at the new location. This iterative process repeats until the organization can no longer improve its performance or when we reach a predetermined number of cycles.

3.2.5. Trust and knowledge transfer

The final aspect of our model considers the role of trust in enabling effective knowledge transfer across the units involved in the ISD process. Since ISD units (i.e., business or IS) may know some portion of the other unit's design configuration elements, they can consider alternatives that alter the other unit's design choices. Continuing with the above example, if $k = 1$ and it was randomly determined that the business unit also knows d_7, then the business unit may consider the alternative configuration (1, 0, 0, 1, 1, 1), which does not alter any of its own design choices (i.e., d_1 through d_5 remain the same), but alters the other unit's design choice (i.e., d_7). If this alternative has a higher fitness level than the current status quo, the business unit can very well request to the IS unit that this change be made (i.e., that d_7 be changed from 0 to 1). If there is trust between the ISD units, then the IS unit will accept this proposal and update its design choice configuration accordingly. In the absence of trust, the IS unit will disregard this proposal and initiate the systems design phase from its existing design choice configuration.

4. ANALYSIS AND EXPERIMENTAL DESIGN

To ensure that the results of the simulation reflect the underlying structure of the model rather than a particular realization of a stochastic process, we use the procedure outlined in the previous section to generate many landscapes with the same underlying pattern of interaction (i.e., a predetermined influence matrix). The results are based on 200 independently generated fitness landscapes (for each influence matrix), with 100 ISD projects randomly seeded onto each fitness landscape.

---

will consider 6 (i.e., $N/2 + k$) neighboring alternatives – {(0, 0, 0, 1, 1, 0), (1, 1, 0, 1, 1, 0), (1, 0, 1, 1, 1, 0), (1, 0, 0, To generate sufficiently complex fitness landscapes for ISD projects, we set $N = 16$ so each unit (business and IS) has, by default, responsibility for 8 design choice configuration.[8] Each simulation run is executed for 100 simulated time periods, by which time most ISD projects have reached a stable state (i.e., additional improvements cannot be found).[9]

We created three sets of influence matrices for comparison across patterns of interactions. First, we varied the K parameter at three levels (i.e., $K = \{0, 7, 15\}$) to represent three levels of overall landscape complexity for $N = 16$. When $K = 0$, each design choice is independent and the resulting landscape is smooth. When $K = 15$, the resulting landscape extremely rugged and search within this landscape extremely difficult as a single change in one design choice changes the value contributions of all other design choices. Finally, $K = 7$ represents the middle ground – a moderately rugged / complex landscape. Second, we varied the patterns of interactions within the moderately rugged landscape setting ($K = 7$) by manipulating the number of interdependencies both within and between units. These landscapes are denoted W7B0 (i.e., 7 interactions within unit and 0 interactions between units), W6B1, W5B2, W4B3, W3B4, W2B5, W1B6 and W0B7. We also created two additional influence matrices that have a skewed distribution of interdependencies while keeping the average K for the landscape fixed at 7. One such matrix was unevenly distributed to favor the upper right block (i.e., business design choices depend on IS design choices more than IS design choices depend on business design choices), and the other to favor the lower left block (i.e., IS design choices depend on business design choices more than business design choices depend on IS design choices). Finally, we varied the parameter k for knowledge overlap by considering all values from 0 (i.e., no knowledge overlap) to 8 (i.e., full knowledge overlap). All models were run with and without *trust* between units.

0, 1, 0), (1, 0, 0, 1, 0, 0), (1, 0, 0, 1, 1, 1)}.

[8] With $N = 16$, there are $2^{16} = 65536$ possible configurations of ISD projects.

[9] Each period represents one phase (requirements determination or systems design) in the ISD cycle. With an incremental / iterative development process, a single phase may take anywhere from a few days to a few weeks. If we consider, an agile development process where requirements determination and systems design iterations are completed on a weekly cycle, 100 time periods represents 100 weeks, or approximately 2 years.

5. RESULTS

Our simulation results are shown and discussed in this section. Due to space limitations, only the most revealing results are presented. Table 3 presents descriptive statistics of ISD performance at time period 15 and at time period 100 under various experimental conditions. Each result is the average normalized performance of 100 ISD projects across 200 landscapes.[10] Most of the results that we articulate below are statistically significant at $p < 0.01$, unless otherwise noted. However, due to space limitations, we do not present detailed statistical analyses including t-tests.[11] For more intuitive interpretations of these results, Figures 3 to 6 are presented below.

Table 3. ISD Performance Results

	Performance at t = 15					Performance at t = 100				
	k = 0	k = 2	k = 4	k = 6	k = 8	k = 0	k = 2	k = 4	k = 6	k = 8
Panel 1. Performance across overall *landscape complexity*										
K = 0	0.927	0.905	0.888	0.871	0.859	1.000	1.000	1.000	1.000	1.000
K = 7	0.730	0.737	0.748	0.763	0.783	0.762	0.793	0.823	0.853	0.881
K = 15	0.642	0.655	0.671	0.702	0.766	0.646	0.665	0.694	0.742	0.808
Panel 2. Performance across *patterns of interactions*										
W7B0	0.818	0.805	0.795	0.789	0.791	0.857	0.855	0.854	0.853	0.857
W6B1	0.758	0.758	0.761	0.767	0.781	0.789	0.809	0.825	0.842	0.865
W5B2	0.727	0.733	0.744	0.760	0.782	0.754	0.784	0.812	0.841	0.871
W4B3	0.706	0.719	0.735	0.756	0.782	0.731	0.767	0.802	0.839	0.873
W3B4	0.697	0.711	0.730	0.753	0.781	0.723	0.758	0.794	0.836	0.871
W2B5	0.705	0.718	0.734	0.754	0.782	0.729	0.766	0.800	0.838	0.872
W1B6	0.728	0.736	0.747	0.762	0.785	0.756	0.785	0.813	0.842	0.874
W0B7	0.762	0.761	0.763	0.772	0.785	0.795	0.813	0.830	0.847	0.870
Panel 3. Performance difference between *Trust* and *No Trust* conditions across patterns of interactions										
W7B0	0.000	-0.013	-0.012	0.003	0.033	0.001	0.000	-0.001	0.008	0.000
W6B1	-0.001	-0.001	0.004	0.015	0.040	0.000	0.003	-0.003	0.013	0.001
W5B2	0.000	0.004	0.011	0.023	0.042	0.000	0.005	0.005	0.015	0.000
W4B3	0.000	0.005	0.013	0.026	0.044	0.001	0.005	0.009	0.017	0.001
W3B4	0.000	0.006	0.014	0.027	0.043	0.001	0.006	0.011	0.017	0.000
W2B5	0.001	0.005	0.014	0.028	0.043	0.001	0.005	0.008	0.016	0.001
W1B6	0.001	0.004	0.009	0.022	0.043	0.000	0.005	0.005	0.016	0.000
W0B7	-0.001	0.000	0.004	0.014	0.040	0.000	0.002	-0.003	0.012	0.001

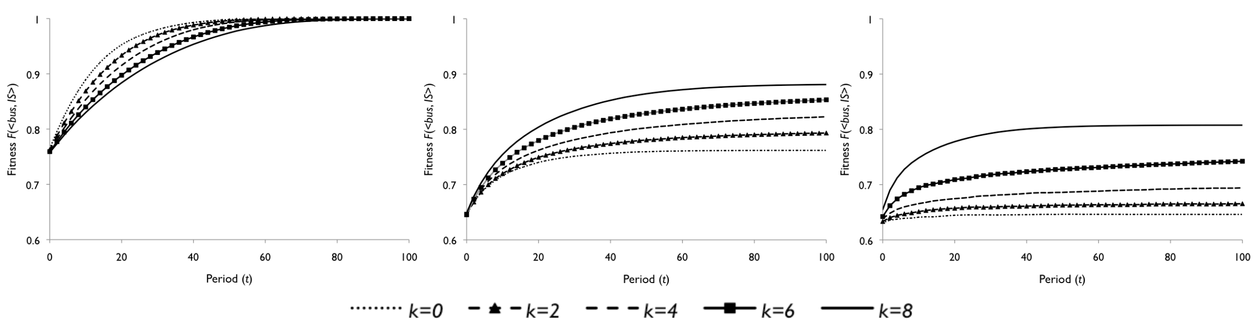
Notes: Each result is an average of 100 projects across 200 landscapes of each type.

5.1. Impact of Interdependence on ISD Performance

Figure 3 shows how different levels of knowledge overlaps ($k = \{0, 2, 4, 6, 8\}$) between business and IS units influence the performance of the ISD project over time when the overall level of

[10] Since each landscape may have a different global optimum value, we normalize the actual performance by the maximum attainable performance for each given landscape generated.

interdependencies is very low ($K = 0$), moderate ($K = 7$), and very high ($K = 15$), assuming inter-unit trust does not exists. The results clearly show that ISD performance decreases as interdependencies increase (i.e., going from $K = 0$ to 7 to 15), regardless of knowledge overlap levels. When interdependencies do not exist (Figure 3a; $K = 0$), all ISD projects eventually reach global optimum (i.e., performance = 1) as can be expected in a smooth fitness landscape. However, we note that the rate at which the global optimum is reached is actually faster with lower knowledge overlap. It seems that the overlapping knowledge between units is inducing the ISD projects to explore a greater number of design configurations, which may not be necessary within a smooth fitness landscape. When interdependencies are moderate (Figure 3b; $K = 7$) or very high (Figure 3c; $K = 15$), ISD projects do not reach global optimum.



(a) N = 16, K = 0 (b) N = 16, K = 7 (c) N = 16, K = 15

Figure 3. ISD Performance over Time with Varying Levels of Overall Interdependencies

The performance gaps among different levels of knowledge overlap are found to be larger when interdependencies are higher. When $K = 0$, we found relatively small performance gaps among different levels of knowledge overlap. However, when $K = 7$ or $K = 15$, performance gaps among different levels of knowledge overlap are larger and become even larger over time. We also note that unlike with the case of the landscape in which interdependencies did not exist, ISD projected endowed with greater knowledge overlap perform better than those with lesser overlap. With more complex landscapes, it seems that the

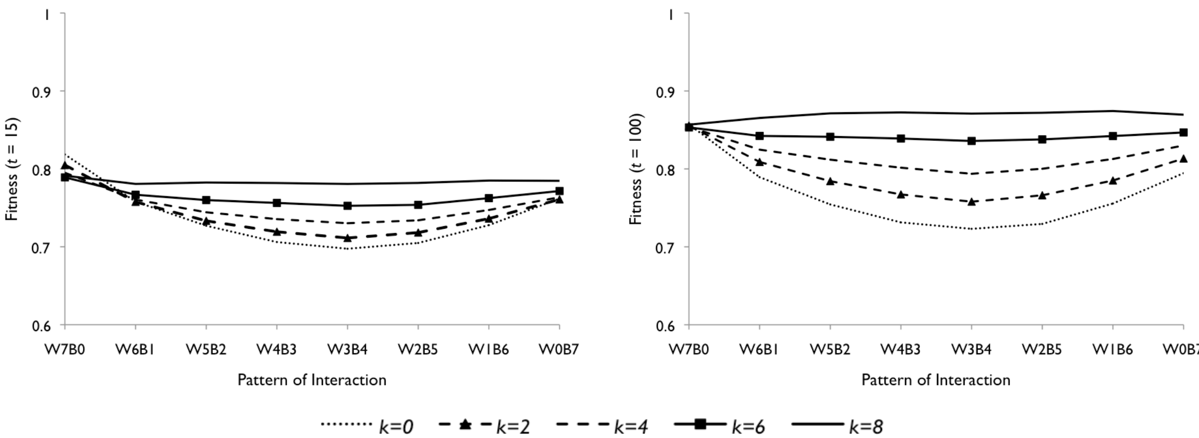
[11] Statistical test results are available from the authors upon request.

additional exploration of alternatives due to the overlapping knowledge across units is helping the ISD project team to better navigate the rugged landscape. In fact, the eventual performance of highly complex ISD projects (i.e., $K = 15$) when the level of knowledge overlap is high (i.e., $k = 8$) is greater than the eventual performance with moderately complex ISD projects (i.e., $K = 7$) when the level of knowledge overlap is low (i.e., $k = 0$ or 2).

When inter-unit trust exists, though the results are not presented here due to space limitations, we find that the ISD projects reach their eventual levels much faster for all levels of interdependencies. However, the actual level of final performance is not significantly different from when inter-unit trust does not exit. We also find that the negative impact of knowledge overlap when inter-unit interdependencies do not exist ($K = 0$) vanishes when inter-unit trust exists.

5.2. Patterns of Interdependence and Knowledge Overlap

Figure 4 shows how different levels of knowledge overlaps ($k = \{0, 2, 4, 6, 8\}$) between business and IS units influence ISD performance at $t = 15$ and $t = 100$ when, holding K constant to 7, the distribution of within-unit interdependencies and between-unit interdependencies varies from W7B0 to W6B1, W5B2, W4B3, W3B4, W2B5, W1B6, and W0B7, assuming inter-unit trust does not exist. The results show that, regardless of the distribution of interdependencies, knowledge overlap is positively associated with ISD performance, except for the case of W7B0 where all knowledge overlaps lead to the same ultimate performance (at $t = 100$). Furthermore, a greater knowledge overlap results in more consistent performance across different distributions of interdependencies. Interestingly, ISD performance tends to be lower when within-unit interdependencies and between-unit interdependencies are evenly distributed (e.g., W3B4) than when those interdependencies are unevenly distributed (e.g., W6B1, W0B7). The only exception is that when a complete knowledge overlap exists (i.e., $k = 8$), performance does not change across different distributions of interdependencies. As a result, the performance gaps among varying levels of knowledge overlap become larger as interdependencies become evenly distributed. Finally, the performance gaps among varying levels of knowledge overlap are larger at period 100 than at period 15.



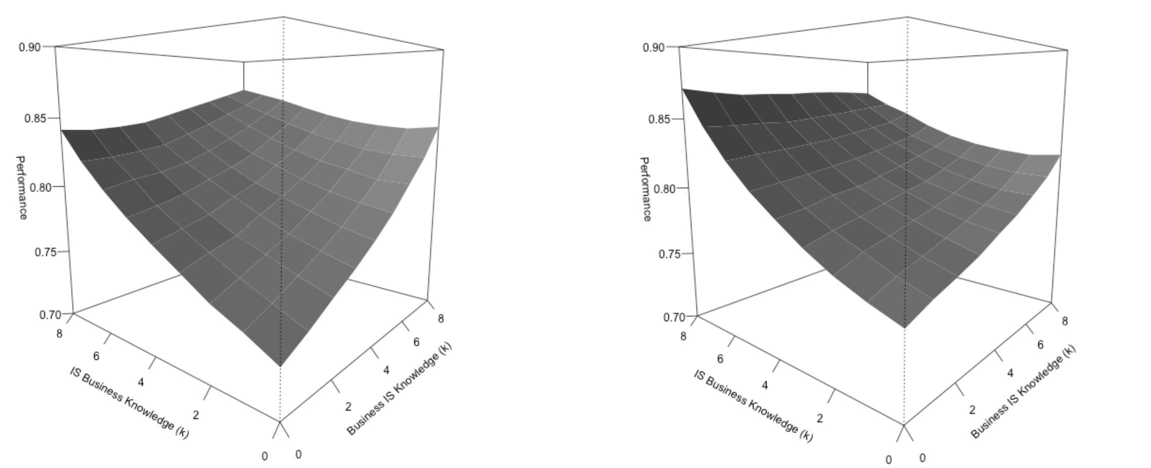
(a) Performance at time period = 15 (b) Performance at time period = 100

Figure 4. ISD Performance with Different Distributions of Within- and Between-Units Interdependencies

The U-shaped results for the performance as within-unit interdependencies decrease (and between-unit interdependencies increase) suggest that the specialization-based task partitioning into business and IS domains is less effective when the within- and between-unit interdependencies are evenly distributed (e.g., W3B4). It is quite intuitive to infer why performance would suffer less when there are greater within-unit interdependencies (i.e., business (IS) design choices depend more on other business (IS) design choices rather than IS (business) design choices; e.g., W6B1). Since there aren't many between-unit interdependencies, each unit can focus on optimizing the configuration for its own domain, since design choices of the other unit will not impact the fitness of its design choices. However, that performance suffers less when there are greater between-unit interdependencies (e.g., W1B6) is less intuitive. In such cases, due to the greater level of between-unit interdependencies, each unit is at the mercy of the other unit's design choices. In other words, rather than taking an active role in searching for the optimal local (i.e., within domain) design configuration, each unit will rely on the other units design choices and make its own design decisions that support the design choices of the other unit. Given that the level of within-unit interdependencies is low, the resulting local landscape will be smoother, and consequently this adaptive decision will be efficient and not required extensive search. However, when within- and between-unit interdependencies are more evenly distributed (i.e., W3B4), search is globally more difficult since each unit must actively search within its rugged local landscape and design choices of the other unit can also drastically alter its own landscape. However, the results suggest that greater knowledge overlap can help to mitigate such difficulties.

5.3. Distribution of Knowledge Overlap

Figure 5 shows how different levels and distributions of knowledge overlaps influence the final performance of the ISD project when the distribution of between-unit interdependencies varies, holding the total amount of interdependencies in the influence matrix constant and assuming inter-unit trust does not exist. Specifically, Figure 5a shows the results when business and IS design choices are evenly interdependent whereas Figure 5b shows the results when IS design choices are more dependent on business design choices than the other way around.



(a) When business and IS design choices are evenly interdependent

(b) When IS design choices are more dependent on business design choices than the other way around

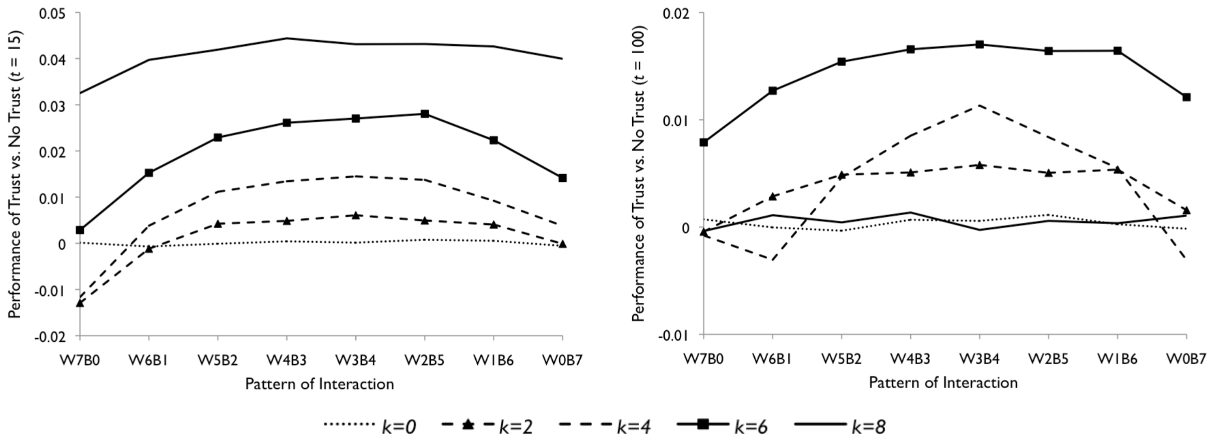
Figure 5. ISD Performance at Period = 100 with Different Distributions of Knowledge Overlaps and Between-Unit Interdependencies

The results suggest that, in general, greater level of knowledge overlap is beneficial – see upward sloping plane in Figure 5a. However, we also note that performance is greater when knowledge overlap is unevenly distributed. For instance, we see that ISD performance is greater when the IS unit's business knowledge is greater (lesser) than the business unit's IS knowledge (e.g., k_{is} = 8 and k_{bus} = 0; k_{is} = 6 and k_{bus} = 2; k_{is} = 2 and k_{bus} = 6 or k_{is} = 2 and k_{bus} = 8), than when they are equal (i.e., k_{is} = k_{bus} = 4), even when the business and IS design choices are evenly interdependent.

ISD performance was also found to be higher when the knowledge overlap pattern matches the distribution of interdependencies. For example, as shown in Figure 5b, when IS design choices are more dependent on business design choices rather than the other way round, business knowledge of the IS unit contributes to performance gain more than IS knowledge of the business unit does. Similarly, we found that when business design choices are more dependent on IS design choices than the other way round, IS knowledge of the business unit contributes to performance gain more than business knowledge of the IS unit does.

5.4. Inter-Unit Trust and Knowledge Overlap

Figure 6 shows how trust between business and IS units influences early and ultimate ISD performance when, holding K constant to 7, the distribution of within-unit interdependencies and between-unit interdependencies varies from W7B0 to W6B1, W5B2, W4B3, W3B4, W2B5, W1B6, and W0B7. More specifically, Figure 6a shows the performance difference between trust and no trust conditions at time t = 15 and Figure 6b shows the performance difference at t = 100. Although the difference in performance between trust and no trust conditions appears to be generally small, ranging from - 0.01 to 0.04, the performance difference tends to be larger at time period 15 than at time period 100. Furthermore, performance difference tends to be larger when interdependencies are evenly distributed than when they are unevenly distributed. When k = 0, performance difference is not found because trust would not matter with no knowledge overlap. When k = 8, early performance with trust at t = 15 is much higher than that with no trust. However, such performance difference in early periods disappears at t = 100 – when a high level of knowledge overlap exists, inter-unit trust is no longer required.



(a) Performance difference (trust - no trust) at time period = 15 (b) Performance difference (trust - no trust) at time period = 100

Figure 6. ISD Performance Difference between Trust and No Trust Conditions

6. DISCUSSION

In this paper, we present a novel conceptualization of the ISD process as search within a design space made up of business domain and technical configurations. This conceptualization allows us to explore the dynamic relationship between the structure of the complexity of ISD projects and the distribution of relevant knowledge across the organization in effecting the efficacy of finding high-quality configurations for information systems. More specifically, our *NK* fitness landscape model of ISD equips us to investigate the effect of knowledge overlaps on ISD performance under various different conditions of level of interdependencies, distribution of interdependencies, and inter-unit trust. This research extends and elaborates on the conventional wisdom that the knowledge overlaps between business and IS units are positively associated with ISD performance. The primary objective of this research was to derive new or nuanced theoretical insights about the effect of knowledge overlap on ISD.

Prior literature has investigated the direct effect of knowledge overlap between the IS unit and the business unit on the performance of the ISD process in terms of coordination and decision control (Espinosa et al. 2007; Tiwana 2009). Furthermore, one study has investigated the moderating effect of project and process novelty on the relationship between knowledge overlap and outsourcing ISD performance (Tiwana 2004). However, prior research has not yet fully revealed the complex conditions under which knowledge overlap affects ISD performance differently. This research fills the gap. We found that the relative effect of knowledge overlap on ISD performance depends on the level of interdependencies of design choices, patterns of such interdependencies, and distribution of overlapping knowledge.

6.1. ISD Complexity

The complexity of an ISD project depends not only on the number of design choice elements but also and more importantly on the extent of interdependencies between design choices. Naturally, complex ISD projects are generally more difficult to conduct than simple ones. The results of the simulation analyses clearly show that as the overall level of interdependencies (K) increases overall performance decreases (see Figure 1). However, we also observe that extent of knowledge overlap across the business and IS units helps deal with ISD complexity and the impact of knowledge overlap on ISD performance is heightened with increasing ISD complexity. The moderating role of knowledge overlap is quite remarkable since eventual performance of highly complex ISD projects (i.e., $K = 15$) when the level of knowledge overlap is high (i.e., $k = 6$ or 8) is even greater than the performance of moderately complex ISD projects (i.e., $K = 7$) when knowledge overlap does not exist (i.e., $k = 0$) or when the level of knowledge overlap is low (i.e., $k = 2$). In other words, organizations may use knowledge overlap as a means to reduce the effective level of ISD complexity. These findings are summarized into the following testable propositions:

Proposition 1: As interdependencies among system design choices increase, ISD performance decreases, regardless of knowledge overlap level.
Proposition 2: As interdependencies among system design choices increase, a higher level of knowledge overlap is associated with higher ISD performance.
Proposition 3: As interdependencies among system design choices increase, the performance gap between a higher level of knowledge overlap and a lower level of knowledge overlap increases.

How does knowledge overlap help mitigate the difficulties of ISD complexity? Our conceptualization of the ISD process as search provides a plausible explanation. Prior work on organizational search (e.g., Gavetti and Levinthal 2000; Rivkin and Siggelkow 2003) has highlighted the importance of balancing exploration and exploitation (March 1991) especially when the landscape to search is rugged and multi-peaked (i.e., when complexity is high). Exploration is important because it provides the variation necessary to escape local peaks and find other fertile regions within landscape. Exploitation is important for organizations to zero in on identifying the location of highest return once a general region has been identified. Knowledge overlap across units provides an implicit mechanism whereby each unit's exploratory search efforts are coordinated toward a unified solution. Although each unit searches its design space independently, knowledge of the other unit's design elements helps to have a more accurate assessment of how its design changes will impact the overall fitness for the ISD project. As a result, when a unit (business or IS) searches its design space, it not only worries about how its changes impact the fitness for its own design choice elements but also considers how these changes impact the fitness with respect to the choices of the other unit. In other words, knowledge overlap impels the independent units to search while being cognizant of the impact that its design choices will have on the other unit's fitness level. In doing so, the overall organization can be guided toward high performance regions within the landscape.

That said, we also observed a negative impact of knowledge overlap in terms of hindering the speed at which the organization converges on its optimal design when complexity is low (see Figure 3a). When there does not exist many interdependencies between design elements (i.e., when complexity is low), each unit's design choices should not significantly impact the fitness of other unit's decisions. . However, having knowledge of and considering the other unit's design elements may create unnecessary variation that derails the other units search trajectories. As a result, the overall search process takes longer.

6.2. Patterns of Interdependencies

While the previous discussion centers on a generalized notion of ISD complexity, our analyses provide additional insights into how the patterns of interdependencies impact the complexity of ISD projects and consequently ISD performance.

Although much of the literature using *NK* landscapes models have assumed a random distribution of interdependencies to model complexity, prior research has emphasized that it is not the overall extent of interdependencies but the patterns of interdependencies that actually determine the complexity of a system (Rivkin and Siggelkow 2007). Our study extends this line of work by investigating patterns of interdependencies when there are multiple sub-units with specialized knowledge. Our overall findings can be summarized into the following testable propositions:

Proposition 4: When all interdependencies are associated with design choices within the same unit and no interdependencies are associated with design choices between business and IS units, knowledge overlap does not impact ISD performance.
Proposition 5: A higher level of knowledge overlap results in more consistent ISD performance across different distributions of interdependencies.
Proposition 6: ISD performance is lower when within-unit interdependencies and between-unit interdependencies are evenly distributed than when those interdependencies are unevenly distributed.
Proposition 7: As interdependencies become evenly distributed across within units and between units, the performance gap between a higher level of knowledge overlap and a lower level of knowledge overlap increases.
Proposition 8: When interdependencies among design choices exist, performance gap between a higher level of knowledge overlap and a lower level knowledge overlap increases over time.

Similar to Rivkin and Siggelkow (2007),[12] our analyses show that even though the overall extent of interdependencies is held constant, the patterns of interdependencies has a significant impact on the overall complexity of the system in terms of how easy (or difficult) it is to search for fertile locations on the landscape. We find that ISD projects are most complex when interdependencies are evenly distributed across within and between knowledge domains (e.g., W4B3 and W3B4 in Figure 4). The flip side is that ISD projects are less complex when interdependencies exist mostly within knowledge domains (e.g., W7B0 and W6B1 in Figure 4). Our results suggest that when multiple units are independently searching for higher performing design configurations, one unit's search activities can result in unreliable assessments if the fitness of its design choices heavily depends on the design choices

[12] Rivkin and Siggelkow (2007) investigated archetypical patterns of interdependencies (e.g., random, centralized, hierarchical, small world, scale free, dependent, etc.) with a single unit that searches within these landscapes. Our study is different in that we have multiple (i.e., two) sub-units of the organization specialized in different domains that collaboratively search the landscape where the patterns of interdependencies relate to how design choices impact the fitness of other design choices within and between specialized domains.

of the other unit. As a result, when interdependencies are evenly distributed both within and between the units' specialized domains, each unit's design choices will alter the sub-landscape for the other unit, which makes it difficult to converge upon a high-performing solution. Conversely, when interdependencies are mostly within each unit's specialized domain (i.e., a modular structure of interdependencies), then while the sub-landscape for each unit may be more complex to search, each unit's search activities will be impacted less by the design choices of the other unit. In other words, in such cases, each unit can independently search for fertile ground without worrying about the consequences of the other unit's search activities. Finally, somewhat surprisingly, we find that ISD projects are also less complex when interdependencies exist mostly across knowledge domains (e.g., W0B7, W1B6 in Figure 4). When interdependencies are mostly across the units' specialized domains, this implies that the sub-landscapes for each unit are relatively smooth (even though the fitness values of these smooth sub-landscapes may change significantly as a consequence of the design choices of the other units). As a result, even though the other unit's design choices may alter a focal unit's sub-landscape, the focal unit's search activities can efficiently find an optimal solution (albeit local).

Again, knowledge overlaps helps overcome the difficulties of search arising from the structural complexity due to the patterns of interdependencies. Knowledge of the other unit's design elements allows a more accurate assessment of how one's own design choices will impact the overall fitness for the ISD project. With a better assessment of how the other unit's design choices impact one's own search, each unit need not be blindsided by the search activities of the other units, even when the patterns of interdependencies create opportunities for radical shifts in local sub-landscapes.

6.3. Distribution of Knowledge Overlap

One of the more interesting results relate to the distribution of knowledge overlap between units. Contrary to conventional wisdom in the IS literature, the analyses suggest that higher performance can be attained when knowledge overlap is unevenly distributed between the two units. In other words, holding the total amount of knowledge overlaps constant, it is better to have one unit know a lot about the other unit's domain rather than have both units equally knowledgeable of each other. This effect was heightened when the patterns of interdependencies mirrored the distribution of knowledge overlaps. Our findings are summarized into the following propositions:

Proposition 9: ISD performance is higher when knowledge overlap is unevenly distributed between business and IS units

Proposition 10: ISD performance is higher when the knowledge overlaps pattern matches the distribution of interdependencies; when IS design choices are more dependent on business design choices than the other way round, business knowledge of the IS unit contributes to performance more than IS knowledge of the business unit does, and vice versa.

It seems that organizations are better off if one of the units takes a leadership role in the ISD process. That said, the question of which unit (i.e., business vs. IS) should take the leadership role depends on the patterns of interdependencies for the ISD project. If both units have equal knowledge of each other's domains, then each unit may presume to know what's best for the other unit. However, despite each unit's best efforts to search the landscape that is concurrently favorable to the other unit, such assertive search behaviors may create excessive variation and lead to each unit attempting to go in different directions. Conversely, if one of the units has relatively more cross-unit knowledge, then this unit can take the role of exploring the overall landscape for high performance regions and the other unit can exploit by seeking the locally optimal position within that region. This interpretation is somewhat consistent with the notion of the structural ambidexterity proposed by Tushman and O'Reilly (1996) in that organizations become more effective when one unit is responsible for exploration and the other unit is responsible for exploitation.

6.4. Inter-Unit Trust

Finally, our analyses confirmed the positive role of inter-unit trust in ISD performance. Inter-unit trust enabled search to be faster and helped deal with complexity arising from the patterns of interdependencies. The results relating to inter-unit trust are summarized into the following testable propositions:

Proposition 10: Trust between business and IS units is positively associated with early ISD *performance.*

Proposition 11: The positive effect of trust on ISD performance decreases over time.

Proposition 12: The positive effect of trust on ISD performance increases as interdependencies become evenly distributed.

The speed effect of trust seemed to be due to the increased breadth of search enabled by inter-unit trust. When inter-unit trust exists, each unit may recommend a design configuration that alters the design choices of the other unit. In other words, a unit can effectively conduct search for the other unit. So when a unit encounters a situation where search within its own sub-landscape will produce an inferior solution to allowing the other unit to continue to search its sub-landscape, inter-unit trust effectively enables a prolonged search of either of the domains. As a result, the organization can explore and converge more quickly to a high performance region at which time local search can identify the (locally) optimal position.

Inter-unit trust also helped deal with the complexities arising from the structure of interdependencies. Recall that search difficulties arise because one unit's design choices may alter the structure of the sub-landscape of the other unit. Inter-unit trust allows a more flexible process where search within one unit's sub-landscape can last until a high-performance region can be found. The stability afforded by the prolonged search minimizes the occurrence of sudden alterations of the sub-landscape, which was the main cause of ISD complexity.

7. CONCLUSION

In conclusion, this research leverages the power of the *NK* fitness landscape modeling method to discover new and nuanced insights about knowledge overlap in ISD and identifies a set of propositions that call for future empirical validation to further advance theory and practice. Our study has a number of implications for ISD theory and practice. Our analyses suggest that much of the difficulties in ISD is due to the complexity of the ISD arising from the structure of the interdependencies within and across business and technical domains. ISD organizations would do well to recognize and identify the structure of knowledge interdependencies of the ISD project and match the ISD team with an appropriate level of knowledge overlap so as to ensure adequate performance. However, the literature has yet to explore the notion of ISD complexity from a structural complexity perspective. For instance, the systems development methodologies practiced in the field (and taught in our classrooms) do not incorporate representations, models or tools to allow the systems analyst to adequately model and understand the interdependencies between business domain and technical knowledge. Also, the de facto iterative development process of analysis and design implicitly induces knowledge specialization and task partitioning. We believe that much work is needed to develop better theories of ISD processes. Our study is a first step in this direction, harnessing the power of the *NK* landscape modeling method.

This study is not without limitations. Despite the advantages of simulation methods in their ability to incorporate complex dynamics without worrying about analytical tractability and their ability to study constructs of interest that may be difficult if not impossible to manipulate in field studies, simulation models are stylized theoretical models of reality that require rigorous validation through empirical testing. That said, we believe that our *NK* landscapes model of ISD has allowed us to provide valuable theoretical insights that can guide further theoretical and empirical research.

REFERENCES

Alavi, M., and Leidner, D.E. "Knowledge management and knowledge management systems: Conceptual foundations and research issues," *MIS Quarterly* (25:1), 2001, pp. 107-136.

Almirall, E., and Casadesus-Masanell, R. "Open versus closed innovation: A model of discovery and divergence," *Academy of Management Review* (35:1), 2010, pp. 27-47.

Attewell, P. "Technology diffusion and organizational learning: The case of business computing," *Organization Science* (3:1), 1992, pp. 1-19.

Bassellier, G., Reich, B.H., and Benbasat, I. "IT competence of business managers: A definition and research model," *Journal of Management Information Systems* (17:4), 2001, pp. 159-182.

Bassellier, G., and Benbasat, I. "Business competence of information technology professionals: Conceptual development and influence on IT-business partnerships," *MIS Quarterly* (28:4), 2004, pp. 673-694.

Bassellier, G., Benbasat, I., and Reich, B.H. "The influence of business managers' IT competence on championing IT," *Information Systems Research* (14:4), 2003, pp. 317-336.

Beck, K. and Andres, C. *Extreme Programming Explained: Embrace Change*, Addison-Wesley: Boston, MA. 2005.

Cohen, M.D., March, J.G., and Olsen, J.P. "A garbage can model of organizational choice," *Administrative Science Quarterly* (17:1), 1972, pp. 1-25.

Cyert, R.M. and March, J.G. *A Behavioral Theory of the Firm*, Prentice Hall: Englewood Cliffs, NJ, 1963.

Darr, E. and Kurtzberg, T. "An investigation of partner similarity dimensions on knowledge transfer," *Organizational Behavior and Human Decision Processes* (82:1), 2000, pp. 28-44.

Davis, J.P., Bingham, C.B., and Eisenhardt, K.M. "Developing theory through simulation methods," *Academy of Management Review* (32:2), 2007, pp. 480-499.

Espinosa, J.A., Slaughter, S.A., Kraut, R.E. and Herbsleb, J.D. "Team knowledge and coordination in geographically distributed software development," *Journal of Management Information Systems* (24:1), 2007, pp. 135-169.

Ethiraj, S.K., and Levinthal, D.A. "Modularity and innovation in complex systems," *Management Science* (50:2), 2004, pp. 159-173.

Ethiraj, S.K., Levinthal, D.A., and Roy, R.R. "The dual role of modularity: Innovation and imitation," *Management Science* (54:5), 2008, pp. 939-955.

Feeny, D.F. and Willcocks, L.P. "Core IS capabilities for exploiting information technology," *Sloan Management Review* (39:2), 1998, pp. 9-21.

Gavetti, G., and Levinthal, D.A. "Looking forward and looking backward: Cognitive and experiential search," *Administrative Science Quarterly* (45:1), 2000, pp. 113-137.

Gefen D. "E-commerce: The role of familiarity and trust," *Omega: The International Journal of Management Science* (28:6), 2000, pp. 725-737.

Kauffman, S.A. "Adaptation on rugged fitness landscapes," In D. Stein (Ed.) *Lectures in the Sciences of Complexity*, Addison-Wesley: Reading, MA, 1989, pp. 527-618.

Kauffman, S.A. *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York, 1993.

Ko, D.-G., Kirsch, L. J., and King, W. R. “Antecedents of knowledge transfer from consultants to clients in enterprise system implementations,” *MIS Quarterly* (29:1), 2005, pp. 59-85.

Lee, G. and Xia, W. “Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility,” *MIS Quarterly* (34:1), 2010, pp. 87-114.

Leveson, N.G. “Software engineering: A look back and a path to the future,” *Communication of the ACM* (40:2), 1997, pp. 129-131.

Levinthal, D.A. “Adaption on rugged landscapes,” *Management Science* (43:7), 1997, pp. 934-950.

March, J.G. “Exploration and exploitation in organizational learning,” *Organization Science* (2:1), 1991, pp. 71-87.

March, J.G., and Simon, H.A. *Organizations*, John Wiley: New York, 1958.

Mayer, R.C., Davis, S. and Schoorman, F.D. “An integrative model of organizational trust,” *Academy of Management Review* (20:3), 1995, pp. 709-734.

Miller, J.H. and Page, S.E. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*, Princeton University Press: Princeton, NJ. 2007.

Nelson, K.M., and Cooprider, J.G. “The contribution of shared knowledge to IS group performance,” *MIS Quarterly* (20:4), 1996, pp. 409–429.

Newell, S., and Swan, J. “Trust and inter-organizational networking,” *Human Relations* (53:10), 2000, pp. 1287-1328.

Paul, D.L. and McDanniel Jr., R.R. “A field study of the effect of interpersonal trust on virtual collaborative relationship performance,” *MIS Quarterly* (28:2), 2004, pp. 183-227.

Reich, B. H. and Benbasat, I. “Factors that influence the social dimension of linkage between business and information technology objectives,” *MIS Quarterly* (24:1), 2000, pp. 81-113.

Ribbers, P.M. and Schoo, K.-C. “Program management and complexity of ERP implementations,” *Engineering Management Journal* (14:2), 2002, pp. 45-52.

Rivkin, J.W., and Siggelkow, N. “Balancing search and stability: Interdependence among elements of organizational design,” *Management Science* (49:3), 2003, pp. 290-311.

Rivkin, J.W., and Siggelkow, N. “Patterned interactions in complex systems: Implications for exploration,” *Management Science* (53:7), 2007, pp. 1068-1085.

Rockart, J.F., Earl, M.J. and Ross, J.W. “Eight imperatives for the new IT organization,” *Sloan Management Review* (38:1), 1996, pp. 43-55.

Sako, M. “Does trust improve business performance?” In C. Lane and R. Bachmann (Eds.) *Trust Within and Between Organizations: Conceptual Issues and Empirical Applications*, Oxford University Press: New York, 1998, pp. 88-117.

Siggelkow, N., and Rivkin, J.W. “Speed and search: Designing organizations for turbulence and complexity,” *Organization Science* (16:2), 2005, pp. 101-122.

Tiwana, A. “Beyond the black box: Knowledge overlaps in software outsourcing,” *IEEE Software* (21:5), 2004, pp. 51-58.

Tiwana, A. “Governance-knowledge fit in systems development projects,” *Information Systems Research* (20:2), 2009, pp. 180-197.

Tushman, M.L. and O'Reilly, C.A. “Ambidextrous organizations: Managing evolutionary and revolutionary change,” *California Management Review* (38:4), 1996, pp. 8-30 von Hippel, E. “Task partitioning: An innovation process variable,” *Research Policy* (19:5), 1990, pp. 407-418.

Wood, R.E. “Task complexity: definition of the construct,” *Organizational Behavior and Human Decision Processes* (37:1), 1986, pp.60-82.

Xia, W. and Lee, G. “Complexity of information systems development projects: conceptualization and measurement development,” *Journal of Management Information Systems* (22:1), 2005, pp. 45-83.