Modeling Complexity in New Product Development: Decisions and Dependencies in Team-Based Design Projects

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Kyle Frank Oyama

May

2013

APPROVAL SHEET

The dissertation

is submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

AUTHOR

The dissertation has been read and approved by the examining committee:

Dr. Gerard Learmonth

Advisor

Dr. Raul Chao

Dr. William Scherer

Dr. Michael Smith

Dr. K. Preston White

Accepted for the School of Engineering and Applied Science:

Dean, School of Engineering and Applied Science

May

2013

Acknowledgements

I am extremely grateful for the guidance, feedback, and support received from my dissertation committee: Dr. Gerard Learmonth, Dr. William Scherer, Dr. Raul Chao, Dr. K. Preston White, and Dr. Michael Smith. Their wisdom and mentoring was instrumental in helping me focus and direct my research along a path that led to new insights that I would certainly not have found on my own.

I also wish to express my sincere thanks to Dr. Jackie Huband and Dr. Katherine Holcomb of the University of Virginia Alliance for Computational Science and Engineering (UVACSE). Their expertise and willingness to serve as a sounding board regarding the development of my computer models and simulations made the completion of this dissertation possible.

I would like to thank my parents for teaching me to work hard, believing in me, and providing me with great examples to follow.

To our children, Shelby and Jacob, thank you for your understanding and encouragement. I pray that each of you is fortunate enough to find your passions and talents in this life and that you will bless the lives of many.

My wife, Michelle, provided me with unfaltering support during this journey called graduate school. She believed in me even when, at times, I did not. I am profoundly grateful for her steadfast love and encouragement. There is no one else with whom I would want to spend an eternity of tomorrows.

Finally, I wish to acknowledge the source of all knowledge and truth, our Heavenly Father. Ultimately, His blessings and divine intervention made it possible for me to learn not only about the topic of my research, but learn more about myself and how to become a better instrument in His hands.

Abstract

New product development (NPD) is a key activity for generating and sustaining a competitive advantage for firms. Many business processes aim for standardization and repeatability (e.g. manufacturing) and can be modeled using "paths" or sequences of events. However, in NPD, the analogy of a "path" is less applicable because there are many interdependent and multi-disciplinary tasks, many of which are novel. This dissertation explores the issue of how to model the NPD process. Particular attention is devoted to examination of the NK model—a popular model of interdependent nodes in a network—and its applicability as a potential framework for modeling NPD activity. First, the assumptions that underpin the NK model are reviewed. Several of these assumptions are not congruent with real world NPD. The NK model is then adapted to account for experimentation costs and exploration of the design space. It is shown how the search for a better product configuration is moderated by these two factors. Next, the NK model is extended for NPD by incorporating two realities of NPD management: 1) generally, knowledge exists regarding whether a dependency between components is complementary or conflicting in nature and 2) the outcome of design changes are uncertain, but not entirely random. It is shown that the nature of dependencies within a system is more important than the sheer number dependencies or interactions in a system. It is also shown that product improvements are less rapid than the original NK model would suggest. Lastly, the organizational structure of an NPD project is analyzed to assess its impact on the modeling of such a project. It is shown that the co-evolutionary nature of multiple, interdependent teams are more appropriately modeled using the NKC model, a variant of the original NK model. It is shown that the degree to which teams are coupled has a significant influence on project outcomes—a result that is not evident with the original NK model. A heuristic that helps to overcome the unstable nature of tightly coupled design teams is proposed. A discussion of the theoretical and managerial implications of the research concludes the dissertation.

TABLE OF CONTENTS

List of Figures

"When a product plan receives senior management approval...it would seem that all product engineering has to do is implement the plan. Were it that simple, the tension and pressure in product development would be greatly diminished. But it is not." (Clark & Fujimoto, 1991)

Chapter 1: Introduction

New Product Development (NPD) is an important activity for firms because successful new products drive growth and offer competitive advantage. However, NPD is often not successful. In fact, there is a substantial body of evidence that suggests NPD is a difficult endeavor. For example, a recent survey of 277 communications firms revealed that half of the firms exceeded their budgets in NPD, while 42 percent of firms indicated that there were delays in launching products. Further, 70 percent of the firms surveyed halted the development of new projects (Morelli & Van den Biggelaar, 2009). Difficulty in NPD extends to other industries such as pharmaceuticals (Gupta, Pawar, & Smart, 2007) and defense (Bar-Yam, 2003). The Government Accountability Office has been highly critical of Department of Defense development projects and issued a report, citing widespread cost and schedule breaches across the Department (United States Government Accountability Office (GAO), 2009).

Considering the evidence which suggests that there is significant room for improvement in NPD, a natural question arises: why these problems? Can we attribute poor results to poor management? Or have goals and objectives been overstated and overpromised during the early phases of projects? Perhaps changing requirements are to blame? One recent argument posits that a fundamental reason why NPD projects are difficult is due to complexity (Bar-Yam, 2003).

A more thorough review of complexity will be undertaken in the literature review of this dissertation, but at this point let us recognize that complex systems exhibit a property of non-linearity. Non-linearity is a characteristic of complexity, but it is the dependencies between parts of a system that lead to complexity and its outward manifestations. In other words, small changes in one part of the system can lead to disproportionate changes in other parts of the system. In turn, this non-linearity often leads to seemingly erratic and unpredictable changes during the NPD process, especially when non-linearity is not acknowledged by managers and engineers. These "surprises" and seemingly erratic perturbations in the development process are often frustrating for those involved in the NPD effort. Thus, while the behavior of the final product itself is (hopefully) not complex—in that it is well understood and predictable in most regimes—the development *process* of the final product is very much a complex system and serves as the fundamental motivation for this research effort.

1.1. Research Questions, Approaches, and Methods

Significant learning is possible when questions are asked and then investigated. The questions that motivated this research and learning stemmed from a notion that many models of new product development fail to account for complexity. The main question was, "Is it possible to use complexity science to inform NPD?" Significant time was initially spent researching and reading about complexity and its central tenets of non-linearity, sensitivity to initial conditions, and emergent behavior, among others. One model of complexity—Kauffman's NK model (Kauffman, 1993)—kept surfacing in the research on complex systems. This led to another question: "Can the NK model be applied to the domain of NPD?" The answer to this question appeared to be "yes", but further research on the NK model revealed that some aspects of the NK model might not be congruent with the contextual realities of NPD. Therefore, we asked, "Are there modifications that could be made to the NK model in order to make it appropriate for the study of NPD dynamics?" Finally, we asked, "What insights for NPD can be gleaned from modeling NPD as a complex system?"

After a thorough review of literature, an initial modeling effort was undertaken to better understand how the NK model works. This initial effort led to new insights regarding the NK model. Building upon this first effort, two additional modeling efforts were undertaken to examine how a deeper understanding of the NK model and NPD could be used to build a better model of NPD within the framework of the NK model.

1.2. Dissertation Outline

What follows is a top-level outline of this dissertation. Chapter 2 provides a review of the extant literature. In the literature review, three intersecting streams of research germane to this dissertation are presented and discussed. First, new product development is discussed, with a particular focus on models of the NPD process. Then, research on complexity science is presented, with some attention given to general properties of complexity, and significant attention devoted to describing Kauffman's NK model of complexity. The third area of research reviewed and presented centers on product and organizational architectures, in the context of new product development. Chapter 3 presents the first modeling effort that was conducted as part of this work. It investigates how costs and experimentation in the NPD environment might be modeled in the NK model framework. This investigation leads to a better understanding of how the NK model can be adapted to model the NPD process. In Chapter 4, the NK model is adapted to model two specific contextual elements of NPD: 1) knowledge of complementary versus conflicting dependencies between parts of an NPD project and 2) uncertainty regarding the outcomes of specific design changes. Chapter 5 models NPD as a co-evolutionary process of interacting work teams. This modeling effort is based on a variant of the NK model known as the NKC model (Kauffman & Johnsen, 1991), and investigates specific questions regarding team size and the number of teams on an NPD project, as well as how the structure of inter and intra team dependencies affect NPD project outcomes. Chapter 6 concludes by outlining several insights from this work that inform NPD theory as well as insights for the practice of NPD strategy and management. Also included in this summary are limitations of the current work and recommendations for future work that could benefit the NPD research literature.

Salient contributions of this work include qualitative and quantitative analyses of the NK model which generate new understanding regarding how the NK model may be extended and adapted to capture the realities of NPD. We also reveal that the NK model, despite its popularity and widespread acceptance in the research community for modeling interacting elements or decisions, is ill-suited for modeling NPD projects in which the work is organized into multiple sub-teams. Instead, we show that a lesser-known variant of the NK model—the NKC model—is actually a more appropriate choice for modeling NPD projects organized around sub-teams. Patterns of troubled NPD projects, such as excessive design iterations and development times, can be effectively simulated using the NKC model which allows us to further understand how such patterns emerge.

"Someone should be studying the whole system, however crudely that has to be done, because no gluing together of partial studies of a complex non-linear system can give a good idea of the behavior of the whole." (Gell-Mann, 1995)

Chapter 2: Literature Review

This chapter outlines the background and the impetus for this research by closely examining the literature. Because the process of developing new products is a cross-functional effort, involving several overlapping disciplines, two areas of literature will be examined in order to properly evaluate and frame the domain space for this research. First, a general review of new product development will be conducted. Then, a related literature of complex systems will be reviewed, with an in depth discussion of the NK model of complexity (Kauffman, 1993). Finally, these two topics will be synthesized into key conclusions, which will lay the foundation for a deeper understanding of how ideas from complexity science can potentially inform NPD modeling and, in turn, the practice of NPD.

2.1. New Product Development

In the most traditional approach to NPD, systems and products are developed in a series of sequential stages in which information gained in one stage is then passed to the next stage in the process (Oakley, 1984). This paradigm results in the proverbial "over-the-wall" syndrome illustrated in Figure 1, which leads to longer product development times and increased number of engineering changes late in the process (Jarratt et al., 2011). A fundamental reason for these problems is implicit in the term "over-the-wall"—lack of teamwork leads departments to not communicate or understand problems encountered by other departments until late in the process, forcing late design changes and schedule delays.

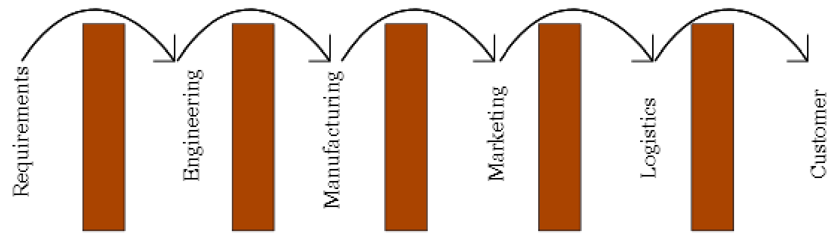

Figure 1. Traditional "over-the-wall" approach to NPD

In an effort to overcome the problems with the traditional sequential approach, the stage-gate approach was introduced (Cooper, 1990). The stage-gate approach (see Figure 2) to NPD is a marked improvement over the traditional sequential NPD process. A significant advantage of the stage-gate process is that, in each stage, information is gathered and shared between functional departments (R&D, marketing, manufacturing, etc) to advance the project to the next phase. During each stage, parallel activities are being worked on by different functional areas. By sharing information between parallel activities early in the process, the goal is to uncover problems earlier in the process and, thus, drive out technical uncertainty sooner, when it is more effective (in terms of cost and schedule) to do so. Additionally, using the stage-gate process, unpromising or poor projects can be rejected in the early stages before more costs accrue.

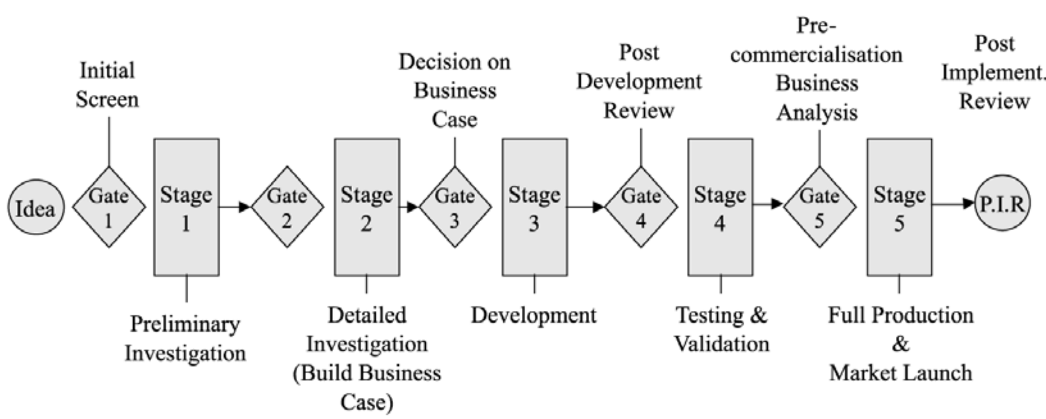

Figure 2. Diagram of Stage-Gate framework for NPD (source: Cooper, 1994)

The stage-gate NPD process has been adopted by many organizations and is widely taught in project management courses because it can help focus an organization's efforts on large, complex projects, and prevent bad projects from proceeding. However, the nature of parallel activities is non-trivial as highlighted in the introduction to this proposal. This is because changes in one activity are subject to uncertainty and can cause changes in other activities which are often disproportionate in magnitude.

A recent trend in the NPD literature directs attention to the decisions that are required to be made during the process of developing a new product or system. Because decision making is not specific to NPD, this literature is multi-disciplinary in nature, borrowing concepts and notions from several fields such as marketing, finance, and engineering. A seminal paper in this literature (Krishnan & Ulrich, 2001) reviewed over 200 papers in the field of NPD and then espoused a view of NPD as a sequence of decision-making. Underscoring the importance of decision-making in NPD, several key studies have examined the psychology of making decisions in a managerial context (Kahneman & Tversky, 2000; Lovallo & Kahneman, 2003; Lovallo & Sibony, 2006). A common theme to these papers concerns managing risk and uncertainty when faced with a decision, yet they also conclude that decision makers, in reality, are often overly optimistic and biased, which can obscure the decision making process and the associated outcomes.

In addition to the stage-gate framework of product development, other frameworks exist which have been influential in shaping how NPD is conceptualized. For example, two of the most popular modeling frameworks that have been applied in the domain of NPD are PERT/CPM[1] (Elmaghraby, 1995) and IDEF[2] (Menzel & Mayer, 2006). However, in an important piece of research on the topic of modeling NPD, it is argued that frameworks such as PERT/CPM and IDEF are more appropriately applied to business processes such as manufacturing rather than NPD (Browning et al., 2006). This argument is made based on the significant differences that exist between NPD and other business processes, such as:

- Most business processes aim for repeatability whereas NPD is characterized by an attempt to create something new, and
- Most business processes are functional in nature while NPD is multi-disciplinary.

Hence, the authors conclude that a network or web is a better way to think of NPD, as opposed to a chain or path. It is further argued that, because NPD processes are characterized by ambiguity, uncertainty, and interdependencies, NPD processes are extremely complex and challenging to model. Finally, the authors conclude that processes can be understood better by the constituent parts *and* interactions in the process (Browning et al., 2006). The notions of networks and interdependencies among parts of a system or process lead us to next examine the related literature of complex systems.

---

[1] PERT: Project Evaluation and Review Technique; CPM: Critical Path Method

[2] Integration Definition

2.2. *Complex Systems*

In Herbert Simon's seminal work, *The Science of the Artificial*, he describes complex systems as being comprised of "a large number of parts that interact in non-simple ways" (Simon, 1969, pg. 195). In Simon's definition, "non-simple" implies that interactions among elements of the system are uncertain and, as a result, often produce surprising, unanticipated outcomes at the system level. These unanticipated outcomes are a hallmark of complex systems, and are often called "emergent behaviors". The notion of emergence suggests that while the behaviors of a given system may be determined and driven by the system's basic elements, some behaviors can only be observed at the system level. Thus, in the study of complex systems it is not sufficient to only study and understand the properties and behaviors of a system's constituent elements.

As an example of a complex system, consider the multitude of interactions in the climate system. The falling rain influences the growth of plants, which subsequently transpire moisture back to the atmosphere. The water vapor in the atmosphere forms clouds which interact with solar radiation that affects the growth of plants. Additionally, temperature variations generate variations in pressure which gives rise to wind; the wind helps regulate the circulation of oceans; and the oceans, in turn, influence the temperature (Rind, 1999). One can imagine many more interactions that exist within the global climate system and the study of any individual aspect of this system—the wind, the rain, the temperature, etc.—cannot give us an understanding of the global climate system.

In addition to natural systems, complexity exists in man-made systems and societies. Stock markets provide a good example of complex systems that have far reaching impacts. The "building blocks" of a stock market are the individual traders. While a large number of traders contribute to the complexity of the system, it is not the sheer number of traders in the market that makes the system complex. Again, the critical issue is the interaction between traders, through the mechanisms of buying and selling securities. And, while we may understand the behavior of how an individual trader behaves, we cannot anticipate the behavior of the overall system due to the many interactions between heterogeneous agents which produce behaviors only seen at the system level.

Another salient characteristic of complex systems is that relationships and interactions are often non-linear in nature. In other words, small changes in one part of the system can have a large impact on the overall system. Sometimes this non-linearity in complex systems is referred to as the "butterfly effect". The term "butterfly effect" is most commonly associated with weather predictions, and is related to the notion of sensitivity to initial conditions. Edward Lorenz, in particular, was fascinated with the study of weather and he developed a mathematical model of the weather. One day, Lorenz wanted to re-examine some of the output from his model, and instead of restarting the entire simulation, he restarted the run in the middle of the simulation, in an effort to save time. However, instead of re-entering the full six digits of data that the computer was using, he used the three digits found on his print out (three digits were used to save space). This seemingly small modification to the data led to very different results in the simulation output (Lorenz, 1995). After verifying that the different results were not a result of a computer malfunction, Lorenz concluded that long-term weather forecasting was going to be a very difficult proposition. And so it is with complex systems that non-linear dynamics make such systems very sensitive to initial conditions (Strogatz, 2001).

In reviewing the literature on complexity, it is important to distinguish what is complex and what is merely complicated, as the terms are often mistakenly interchanged. In complicated systems, a large number of parts interact in predictable ways, and redundancies are built in to the system to handle failures of individual components. For example, an automobile or a modern day jetliner contains thousands, if not millions, of parts but the system behaves in predictable ways (thankfully!) under almost all operating conditions. So, it is not the *number* of parts that makes a system complex. Instead, in complex systems it is the non-simple interactions between the parts that gives rise to complexity (Ottino, 2003).

Regarding the parts of complex systems, the individual building blocks of a complex system—the components, elements, or agents—are able to sense and exchange stimuli with one another, and adapt to their environment. This ability to sense and react leads to a property of complex systems referred to as adaptability. As agents make choices, their actions have an influence on other agents in the system who, in turn, adapt their behaviors and actions in response to their altered environment (Holland, 1996). Thus, the behavior of complex systems is also evolutionary in that the system has a history which is predicated on the past behaviors of agents in the system. An entire literature on complex adaptive systems has developed and grown in popularity over the past few decades (Holland, 1996). The study of social-ecological systems (Ostrom, 2009), for instance, draws upon the ideas of complex adaptive systems to examine some of the world's most challenging problems such as sustainability and how decisions made by many actors or agents bear upon this challenge.

The evolutionary and adaptive nature of biological systems served as the motivation for what has become one of the most popular models of complexity today—the NK model (Kauffman, 1993). The foundation for investigation into NPD in this dissertation draws heavily on Kauffman's NK model and, therefore, this model is now described in more detail.

2.3. *Kauffman's NK Model*

Stuart Kauffman, a theoretical biologist, developed a model to investigate the problem of DNA sequence evolution. In *The Origins of Order* (Kauffman, 1993), Kauffman outlines the NK model which has two primary features. The first feature is a stochastically generated fitness landscape, on which "higher peaks" correspond to better solutions or combinations of elements. The second feature is the agent(s) that search a given landscape in an effort to improve their "fitness" (performance).

Constructing the Landscape. In the model, Kauffman uses two parameters, N and K, to generate the fitness landscape. The parameter N represents the number of components in a given system and is characterized as a string of N binary digits.[3] The value of each digit in the string describes a specific variant of each component. Thus, for a system comprised of N components, the number of possible configurations of the system grows exponentially in N as:

$$\#\ of\ configurations = 2^N$$

The second parameter in the model, K, describes the degree of interdependence between the components of a system. Specifically, K represents the number of other components in the system that affect the effectiveness or fitness value of each component. Thus, the value of K can range from 0 to N-1 and each component can assume randomly drawn fitness values. Figure 3 shows the binary string representation of an example system (N=4) in which each component is affected by 1 other component in the system (K=1). Each component has a value of either 0 or 1, indicating its state or variant, and the directed arrows show the "epistatic"[4] or dependent interactions between components.

[3] Each digit can theoretically assume A possible "alleles" but Kauffman restricts his analysis without loss of generality to systems in which A=2, or systems that can be described as a binary string of N digits.
[4] Epistasis is a term from the field of genetics and refers to a phenomenon in which expression of one gene depends on the presence of one or more "modifier genes."

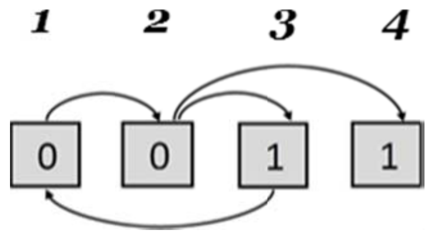

Figure 3. Example of a 4-component system with each component affected by 1 other component

For instance, the first component affects the function of the second component, while the second component affects the function of the third and fourth component, and so on.

Another way to conceptualize interactions between components in complex systems is a matrix representation, similar to the design structure matrix (DSM), introduced in the NPD literature by Steward (1981). Figure 4 illustrates our example system, this time represented in matrix form, where an "X" in the matrix indicates the fitness of the row component is dependent upon the fitness of the column component.

Component	1	2	3	4
1	X		X	
2	X	X		
3		X	X	
4		X		X

Figure 4. Dependency matrix representation of an example system.

The pattern of interactions between components in the system is typically random in the NK model, which stems from the notion that the interaction effect between components in complex systems is often unknown (Simon, 1962). In Figure 4, note the row sums are equal to K+1 (2 in this case). However, the column sums are not necessarily equal to K+1, implying, in the general case, that some components can be more connected and influential than other components.

Continuing with this example, a fitness landscape can be constructed given parameters N and K, and the dependency matrix. As described by Kauffman, each component "mutation" (change in state) results in a new component fitness value, f_c, drawn from the standard uniform distribution for itself and any components which it affects. For instance, a mutation in the second component in our example system would result in new, randomly assigned fitness values for itself, as well as the third and fourth components. Overall fitness, F, for a system is given by the arithmetic mean of all component fitness values in the system:

$$F = \frac{1}{N}\sum_{c=1}^{N} f_c \tag{1}$$

Table 1 illustrates an example of all fitness values that could result from the epistatic interdependencies for all possible configurations of component states.

Configuration	Component Fitness				Overall Fitness
0000	0.2769	0.0462	0.0971	0.8235	0.3109
0001	0.2769	0.0462	0.0971	0.0344	0.1137
0010	0.6948	0.0462	0.9502	0.8235	0.6287
0011	0.6948	0.0462	0.9502	0.0344	0.4314
0100	0.2769	0.3171	0.7655	0.7952	0.5387
0101	0.2769	0.3171	0.7655	0.6463	0.5015
0110	0.6948	0.3171	0.4456	0.7952	0.5632
0111	0.6948	0.3171	0.4456	0.6463	0.5260
1000	0.4387	0.3816	0.0971	0.8235	0.4352
1001	0.4387	0.3816	0.0971	0.0344	0.2380
1010	0.1869	0.3816	0.9502	0.8235	0.5856
1011	0.1869	0.3816	0.9502	0.0344	0.3883
1100	0.4387	0.4898	0.7655	0.7952	0.6223
1101	0.4387	0.4898	0.7655	0.6463	0.5851
1110	0.1869	0.4898	0.4456	0.7952	0.4794
1111	0.1869	0.4898	0.4456	0.6463	0.4422

Table 1. A sample instance of a stochastic fitness landscape for the example system described in Figures 3 and 4.

Searching the Landscape. The second salient feature of the NK model involves an agent(s) searching on fitness landscapes such as those described above. An agent searching a given fitness landscape does so using trial-and-error, in which one component is mutated at random. If, after the mutation, the agent finds the new system fitness value is greater than its current fitness value, the agent "moves" to the new configuration (position on the landscape). If the mutation does not yield a higher system fitness value, the agent retains its previous configuration and mutates a different randomly selected component. Search stops when no greater fitness values are available by mutating one component in the system. Simon notes that this type of local trial-and-error search is congruent with how biological evolution and selection occurs through mutation (Simon, 1969).

Kauffman's original intent in developing the NK model was to build a model of complexity that could be tuned with a single parameter. The parameter K, the degree of interaction between components, is the parameter used to adjust the "shape" or "topography" of the landscape. In the extreme case, where K=0, each component is independent of all other components in the system and, thus, there is one globally optimal configuration of components. However, the search becomes non-trivial for values of $K > 1$, as searches will often terminate at a local optimum due to changes in one bit causing changes in K other bits which frustrate the search.

For example, in Table 1, the string "1100" is a local optimum because each of its "one-mutant neighbors" is of a lower fitness value. Thus, in a context of organizational decision making, one can see how this model of complexity using the K parameter could model conflicting constraints and the propagation of changes to other parts of the organization or system.

In addition, by generating a large number of simulated fitness landscapes, and adjusting the K value, Kauffman also found the number of local optima in the landscape increases exponentially in K. Thus, for higher values of K, the landscape is characterized as being more "rugged" and searching agents will terminate their search at a local optimum with greater frequency. Perhaps most significantly, Kauffman found that the highest mean fitness values for systems of size $N > 8$ tended to exist in systems where K was approximately equal to 3, even as N increased to larger and larger sizes. This implies that there are advantages at a moderately low level of complexity. In other words, complete system decomposition was not found to be optimal, and complete integration of system components (K=N-1) leads to a "complexity catastrophe". A complexity catastrophe is the effect of higher complexity (K) leading to increasingly rugged landscapes with local peaks that proliferate in number and are less differentiated from the overall landscape (Kauffman, 1993).

For a more thorough treatment of Kauffman's NK model, the reader is referred to additional research (Altenberg, 1997; Frenken, 2001) in which statistical properties of the landscapes, algorithms for generating landscapes, and search algorithms are discussed in more detail.

2.3.1. NK Model Applications in Organizations and Strategy

Levinthal (1997) is often credited with introducing the NK model to the field of organizational strategy. He posits that organizational adaptation and population level selection are interrelated and he draws upon the McKinsey 7S framework (Waterman et al., 1980) to argue that mutually reinforcing policies do exist within organizations. Citing literature in the field of evolutionary economics, Levinthal also notes that Milgrom and Roberts (1990) explored the significance of complementary interactions when deciding upon product lines, manufacturing strategies and technology. Levinthal then uses the NK model to examine two processes of change in organizational form, using the N parameter to define the number of attributes in the organizational form and the *K* parameter to define the degree that the fitness contributions of a given attribute of organizational form depends on other attributes. The two processes examined are 1) *local search* (incremental change) as described by Simon (1969) and Kauffman (1993) and 2) *long jumps* (radical change) in which organizational attributes (components) can be mutated (changed) more than one at a time. Modeling organizations using the NK model, in which organizations employ local search *and* long jumps, Levinthal reported findings at the organizational (individual firm) level and the industry (population) level that have been further explored and extended by other researchers.

For individual firms, Levinthal (1997) found that local search is path dependent . In other words, the current state, or configuration, of an organization is largely dependent upon its previous state(s), because only one element of the organization is changed at a time. This is congruent with the notion of local optima in Kauffman's NK model because the local optimum found is highly dependent upon the starting position of the organization. Eisenhardt's (1988) empirical research on institutional theory lends support to this finding, in that the conditions in place at an organization's inception were found to highly influence the decisions made regarding compensation policies. Other researchers (e.g. Anderson, 1999; Dooley, 1997) have noted that a hallmark of complex systems is their sensitivity to initial conditions. Thus, the sensitivity of organizational performance to initial decisions or resources is important in characterizing organizations as complex adaptive systems. Complex adaptive systems are a class of complex systems in which interacting agents learn and adapt. Most organizations, thus, can be thought of as complex adaptive systems due to being composed of agents who learn and adapt based on their experiences and interactions.

The value of *K* in the NK model indicates the degree of interdependency, or coupling, between components in a given system. For organizational modelers, the *K* parameter not only tunes the landscape itself, but has implications for the agents searching the landscape and overall industry dynamics. Similar to Kauffman's assertion (1993), Levinthal (1997) argues that in situations when the value of *K* is high, a higher number of organizational forms emerge due to the presence of more local optima in the landscape. Especially interesting is Levinthal's examination of changing fitness landscapes which were implemented by re-specifying the landscape midway through each simulation run. In these dynamic environments, high values of *K* were also found to lead to reduced survival rates for organizations because tight coupling of elements (high numbers of epistatic interactions) causes difficulty in performing local search due to the rugged landscape. This implies that organizations may need to experiment with long jumps (radical change) as well as local adaptation in order to survive in dynamic environments as noted by Brown and Eisenhardt (1998). However, finding a balance between incremental improvement (exploitation of current opportunities) and radical change (exploration of future alternatives) is not a trivial matter (March, 1991; Tushman & O'Reilly, 1996). What the NK model does show, however, is that in complex environments (high values of *K*), where decision dependencies are numerous, search efforts for improvement using incremental change are exhausted relatively quickly and more radical search becomes critical for escaping local optima (Kauffman, 1993; Levinthal, 1997).

Kauffman's specification of the NK model provided for random assignment of the *K* interactions of equal strength between components of the system. Two sets of organizational researchers, using the NK model, modified this assumption to explore the effects of decision interaction structures. First, Ghemawat and Levinthal (2008) hypothesized that some choices organizations make may be more strategic than others and that there are temporal connections between choices. Specifically, they examined the effect of how components (choices) interact when not all components (choices) have the same number of connections to other components (choices). By then presetting or "seeding" varying numbers of the more highly connected (more strategic) components to their globally optimal fitness values, they investigated the importance and effectiveness of strategic planning. They found that as a greater number of strategic choices were set to their global optimum, the fitness of the system increased, underscoring the importance of strategic planning. But, the difficulty of strategic planning in a complex environment was revealed in that the fitness value of the system did not approach the global optimum until almost all components (choices) were set to their global optimum. Ghemawat and Levinthal (2008) then investigated the notion of historical constraints, by seeding choices which vary in their degree of interaction, with fitness values distant from the global optimum. This relates back to Levinthal's (1997) argument of path dependence. However, when the interdependencies between components are not equal, and explicit attention was focused on initial choices, it was found that the deleterious effect of even one moderately connected component with a poor initial value was profound (Ghemawat & Levinthal, 2008).

Second, Rivkin and Siggelkow (2007) modified the NK model to explore how different patterns of interaction within firms affect the search for better decision configurations within the context of firms. Holding the total number of interactions fixed, they examine 10 different stylized interaction patterns such as a *block-diagonal* pattern in which each component interacts with *K* other components within its own block and there are no interactions between the blocks. Another pattern of interaction they studied was a *hierarchical* pattern in which components are rank-ordered in terms of the components they affect. Thus, some components are highly influential, in that they influence all other components below them, while others affect a small number of other components or none at all. The block diagonal and hierarchical interaction patterns are graphically depicted in Figure 5.

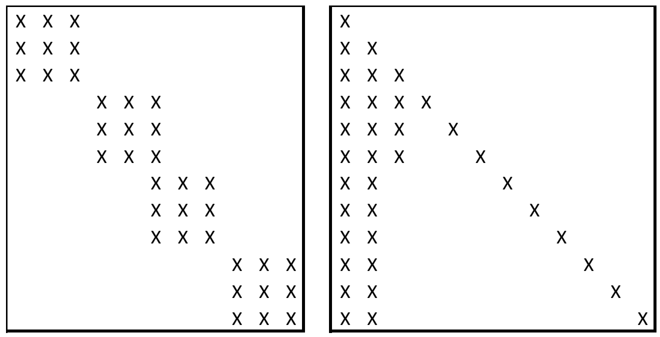

Figure 5. Block-diagonal and hierarchical patterns of interaction.

Rivkin and Siggelkow's (2007) research on patterns of interaction in complex systems made an important contribution with an implication for organizations: the number of interactions is not the only factor that affects the shape of the landscape on which organizations search and the consequent measures of fitness and search length. Rather, the distribution of the interactions plays an important role in defining the landscape topography and can significantly affect the performance of an organization's search. Specifically, they found that when a system or organization has a handful of highly sensitive decisions (that is, they are influenced by changes to a large number of other decisions in the system), then the landscape becomes very rugged and, thus, local search is less likely to be successful. On the other hand, in patterns such as the hierarchical pattern, in which there are many uninfluenced decisions, the landscape is characterized by significantly fewer local peaks, meaning that local search will have a greater chance of continuing before becoming trapped on a local peak.

Thus far, the discussion of NK landscapes and agents has primarily focused on the individual firm. Referring back to Levinthal's initial research (1997) on the NK model applied to organizations, he also found that search and adaptation leads to a few dominant organizational forms (the highest local optima), when combined with selection forces. This is congruent with Kauffman's (1993) argument in theoretical biology that combines self-organization through epistatic interactions with selection forces to yield evolutionary patterns in populations. Similar to population dynamics from biology, Lenox et al. (2007) use the NK model, combined with a Cournot model of competition, to explore the interdependency of decisions in value-creating activities at the industry level of analysis. Specifically, they contend that varying levels of interdependency across industries can explain differing patterns of firm entry and exit, as well as widely varying patterns of industry shakeout. For industries characterized by complexity (higher levels of interdependency in activities), a lower incumbent survival rate and a higher entry and exit rate was found. This result stems from the higher number of local optima that are found, but these optima have little advantage over other local optima and, therefore, the advantage of survival is not sustainable over the long run. In contrast, in industries with low interdependency between activities, there are few local optima and, thus, most competitors end up with very similar products / services that are essentially "commoditized". As a result, entry and exit is slower, turnover is less severe, and significant advantage is afforded to survivors in this type of industry. McKelvey (1999) reports a similar finding that high levels of interdependent activities significantly undermine competitive advantage and the ability to adapt. Lenox et al. (2007) also examined and reported on the sources of innovation across industries with differing levels of interdependency in value-creating activities, finding that in industries with high interdependency, innovation typically comes from new entrants to the market. Conversely, because of the less severe shakeout in industries with lower interdependency, incumbents, more often than not, are the source of innovation.

2.3.2. NK Model Applications in New Product Development

Similar to the preceding discussion of organizations as complex systems, a smaller body of related research using the NK model has focused on exploring new product development (NPD) as a complex set of decisions. Just as the complexity of coordination regarding strategic decisions regarding organizational form increases when large numbers of interacting choices exist (Ghemawat & Levinthal, 2008), Mihm et al. (2003) examine complex product design in the context of distributed decision making in which engineers make many autonomous decisions to optimize the component for which they are responsible, and the fact that these decisions do, in fact, interact and influence other components. Autonomy and interaction between decisions lead to oscillation and engineering change iteration, a key characteristic in the product development activity. This is because the system interactions lead to iterations even if each decision maker, or designer, finds its own best design quickly. In their NK model application to NPD, each individual engineer optimizes their own local performance measure, but interdependencies cause the performance measures of other components to change—the number of interdependencies being governed by the parameter *K*. The contrast of myopic versus cooperative optimization is explored, and simulations of distributed decision making found that a global optimization perspective of designers led to better and faster design solutions (Mihm et al., 2003). Providing the appropriate incentives for individuals or groups to act in this manner, however, is not trivial and has been discussed by Rivkin and Siggelkow (2002) in the context of organizations and fitness landscapes. Therefore subdivision of a product's architecture into nearly independent subsystems has been employed in environments where communication and system level optimization perspectives lead to nonlinear feedback between interdependent designers, design teams, and the components they design. Modularity is one proposed way to reduce the effective size of the problem space and has also been discussed as loose coupling between subsystems in a system's architecture (Maier & Rechtin, 2000).

In another line of research at the intersection of strategy and product development, Chao and Kavadias (2008) investigate strategic decisions regarding resource allocation between incremental and radical NPD projects. These allocation decisions regarding resources is a paradigm called "strategic buckets" in which firms make deliberate splits of resources across various dimensions, one of the most popular of which is the *type of development* dimension (Cooper et al., 1999). Using the NK model to explore this allocation decision in different NPD environments, it was found that in complex environments, characterized by many interdependencies within a product, incremental NPD projects are advantageous because of the higher probability of improved performance, but only in the short term. In the long term, however, radical NPD can escape local optima and lead to even better solutions, provided that there is sufficient time to improve performance between market shifts, which are modeled in a similar fashion to Levinthal's (1997) shifting fitness landscapes. This tradeoff between short-term versus long-term payoffs is also discussed as an important concept when modeling and measuring performance in NK fitness landscapes (Ganco & Hoetker, 2009).

McCarthy et al. (2006) give explicit treatment to describing NPD as a complex adaptive system. They use a case study methodology to elicit major characteristics of complex adaptive systems that are congruent with NPD processes. Their empirical observations revealed non-linearity in NPD, as well as self-organization and emergence. Non-linearity in the context of NPD can manifest itself as minor changes in one subsystem generating disproportionate levels of required redesign in other subsystems. To extend their research, they conclude that a possible future direction for modeling the structural and temporal dimensions of NPD processes could be aided by the use of the NK model. Thus, while some researchers have employed the NK model in their research on NPD and engineering design, it appears that it is a relatively nascent field and that there exists an opportunity to further explore NPD processes using the NK model or modifications thereof.

2.4. Summary of Literature Review

This chapter has examined two streams of literature: new product development (NPD) and complex systems. It is clear from the literature that NPD is a process, characterized by novelty, decisions, and interdependencies. The notion of interdependent decisions made under uncertainty suggests that existing linear process models of NPD may not be sufficient and, instead, a different modeling framework is required (Browning et al., 2006). We also examined the characteristics of complex systems: *emergence, non-linearity, sensitivity to initial conditions, non-simple interactions, evolution, and adaptability.* When we juxtapose the NPD process with the characteristics of complex systems, we conclude that the NPD process is a complex system.

Working under the premise that the NPD process is a complex system, we examined, in some detail, a particular model of complexity—the NK model. On its surface, the NK model appears particularly suited to the modeling of NPD activities. Specifically, the NK model explicitly models a process of search for better configurations of decisions, and NPD is not unlike this type of search activity. Further, this search is conducted on a landscape which is governed by the degree of interdependency between constituent elements of the system which, again, bears a strong resemblance to the interdependency found within NPD projects. Finally, in the NK model, search is conducted by autonomous agents that are generally modeled as being motivated to improve their own performance, or the performance of the element for which they have responsibility. Different agent behaviors, representative of real or hypothesized behaviors in the real world, can and have been implemented and studied in the NK modeling framework.

The NK model, thus, appears to have promise as an appropriate framework for modeling the NPD process. The remainder of this dissertation explores modifications to the NK model that may improve its suitability for NPD modeling, given the particular context in which NPD is situated.

Chapter 3: Extending the NK Model: Part 1

3.1. Introduction

New product development (NPD) projects are a good example of a complex sociotechnical system. For instance, consider the development of a new aircraft. Such a project typically involves hundreds of engineers and managers working in many teams, often organized in a structure similar to the structure of the subsystems of the product itself—e.g. fuselage, flight controls, wing assembly, etc. The coordination of these varied teams' efforts can be conceptualized as a complex system in that individual agents (whether they are actual individuals or subunits such as teams) make decisions and interact with other agents during the development process. Agent decisions not only affect their own subsystem's design but very often also influence the relative performance of other subsystems due to the interactions and dependencies between subsystems. Thus, the development efforts of each subsystem are at the same time variable and interdependent. This coexistence of variation and dependency is a hallmark of complex systems across many domains (Shalizi, 2006). And, therefore, NPD—widely recognized as a source of competitive advantage—can be conceptualized as a complex system. Further, the evolution of the design process in large NPD projects is typically an iterative process of frequent, and often irregular, design changes resulting from new/updated information that becomes available as subsystems are modified. These "surprises" and seemingly erratic perturbations in the development process are often frustrating for system-integrators and managers. Thus, while the behavior of the final product itself is (hopefully) not complex—in that it is well understood and predictable in most regimes—the development of the final product is very much a complex system and serves as the fundamental motivation for this chapter.

Many complex systems are biological in nature, and in this chapter we extend to the NPD domain the NK model (Kauffman, 1993), a model originally developed to study how interdependency between genes in species lead to patterns of self-organization in an overall population. The model uses a parameter, *N*, to describe the number of nodes in a system, and a parameter, *K*, to describe the number of interdependencies of each node. During the roughly two decades since Kauffman published the NK model, management and strategy researchers have employed the NK model in various ways to explore questions regarding interdependent decisions, industry dynamics, organizational design, and product development. One long standing by-product of the industrial revolution was a general conceptualization of organizations as linear "machines" that could be controlled and predicted (McMaster, 1996). However, management theory recognizes organizations as complex, dynamic systems (e.g. Forrester, 1958; Galbraith, 1973; Von Bertalanffy, 1968; Wiener, 1948). Thus, the NK model is particularly attractive to management researchers because it provides a tractable means by which to investigate how different levels of interdependency between parts of an organization affect overall performance of the organization.

In this chapter, we model explore how the NK model may be applied to the domain of new product development as a complex system. Specifically, key assumptions in Kauffman's NK model are challenged and modified to account for dynamics of how novel, large scale, resource intensive systems are developed. Results are presented, followed by a discussion of implications for individuals and organizations involved in development of large-scale complex systems. From a modeling perspective, we find two key dimensions along which the NK model can be extended in order to more accurately model the contextual realities of NPD: 1) the difficulty and costs associated with reversing design decisions in large projects and 2) varying levels of willingness to explore the landscape of possible alternative solutions, despite unpromising initial results, in order to possibly escape local optima. From a more practical managerial perspective, this study finds that as complexity increases in a project, as a result of increased dependencies, the increased costs associated with exploration become have a more pronounced payoff. The investigation and results of this study are a first step in better understanding how the NK model can be adapted and applied to better understand and manage NPD projects in the face of complexity.

The remainder of this chapter is organized as follows: in Section 3.2 we briefly discuss a few of the underlying assumptions in the NK model and contrast those assumptions with the realities of

NPD. Then, in Section 3.3 we present a model of NPD in which we relax and modify some of the assumptions of the original NK model and in Section 3.4 we present the results of our model simulations. Section 3.5 concludes with a discussion of the implications of this research for theory and practice.

3.2. NK Model Assumptions

It should be noted that previous research articles employing the NK model make two critical assumptions that are likely to be unrealistic in the context of innovation and new product development. First, it is assumed agents can explore possible next-steps with perfect knowledge (*and* at no cost) before "moving". This is somewhat unrealistic, however, in innovation and new product development owing to the novelty of the product being developed and the inability to develop perfect prototypes that can be tested in fully realistic environments. Additionally, large scale product development efforts are also typically characterized by largely irreversible capital investments (Dixit & Pindyck, 1994; Ghemawat & Levinthal, 2008). That is, it is not trivial to investigate a possible direction and then decide to abandon that direction if it does not look promising. Second, agents are assumed to only move in directions that result in higher levels of fitness. In the language of new product development, this assumes that organizations do not pursue any unpromising directions, even though they may be able to find a "higher peak" if they learn about the landscape through experimentation, and find that by traversing a "valley" they can find a "higher peak".

3.3. An Extended NK Model for NPD

To explore the nature of decision making in the context of innovative new product development, Kauffman's NK model is used as a starting point. A simulation model of the innovation search and product development activity for a new product is represented using a binary bit string of length 12 (N=12), each bit representing a decision regarding an attribute of a new product. However, to examine the performance of the system when the search algorithm does not allow for comparison of all "next steps", the assumption that fitness must improve at each step in the search is necessarily relaxed. Relaxation of this assumption serves two purposes. First, it incorporates into the model the nature of largely irreversible decisions found in the development of novel new products. In other words, it operationalizes the fact that decision makers are not necessarily able to foresee how their decision(s) today will impact future performance of the system before a decision is made. And, when the monetary costs of implementing a decision are high, it is not trivial to "go back" to the previous point to continue the search. It should be noted, at this point, that the decisions / "moves" in this search are not necessarily irreversible, but rather that any reversal comes at a cost and, thus, the new configuration serves as the de facto new starting position for further exploration.

Second, by relaxing the assumption that fitness must improve at each step, it becomes possible to avoid being trapped on local optima. By accepting the fact that not each step will result in improved system fitness, a system design may be found that is better than the nearest local optima. Of course, this comes at some risk, in that a better solution may not be discovered and, thus, limited time and resources may be expended without any realized benefit. A parameter for this risk seeking behavior is added to the agent search algorithm that, in effect, puts a tolerance on the number of total "steps" the agent is willing to take that do not yield fitness value improvement.

The following questions are examined regarding system performance under conditions of complexity and largely irreversible decision making:

1. How much does it cost to find a satisfactory system design?
2. What is the average fitness value of the final system design?
3. How does the search behavior described above compare to the local-search described by Kauffman?

To explore these questions, the standard local search algorithm for NK fitness landscapes is modified and the following algorithm and agent behavior is implemented:

1. A randomly chosen system configuration, serves as the starting design on a stochastically generated NK fitness landscape
2. At random, one system element (one bit in the binary string) is modified
3. The chosen modification is implemented at a cost modeled by a random variable, normally distributed[5] with mean = 1, standard deviation = 0.2
4. If the new system fitness value is greater than the previous fitness value, search continues from its current location. Otherwise:
    a. The "risk tolerance" counter is incremented and search continues from the new

[5] Cost is modeled as a normal random variable due to lack of any empirical data to suggest otherwise. Future research could involve modeling cost using other distributions—e.g. Beta, Triangular, etc—if data / literature suggest that a particular distribution is more appropriate. It should also be noted that in the rare event that a negative value is sampled from the normal distribution for the cost variable (a case that does not have a physical interpretation that makes sense), a new random variable is sampled and used instead.

current position (which is lower than its previous position)

b. If the "risk tolerance" counter reaches the user defined risk tolerance, the system design reverts back to the system design that had previously yielded the highest fitness value at a cost modeled by a random variable, normally distributed with mean = 0.5, standard deviation = 0.1 (It is assumed, here, that there is a rather significant cost to revert to a previous design.)

5. Once the risk tolerance parameter is exceeded, the search continues as a local search with the following procedure:
   a. The agent randomly selects one system element to modify
   b. The agent implements the chosen modification at a cost modeled by a random variable, normally distributed with mean = 1, standard deviation = 0.2
   c. If the new system fitness value is greater than the previous fitness value, the agent continues search from its current location. Otherwise, the agent returns to previous system design at a cost modeled by a random variable, normally distributed with mean = 0.5, standard deviation = 0.1
   d. Search continues until no further improvement can be made via one-mutant neighbors
6. Record system final system fitness value and total cost

3.4. Results

Six hundred iterations of the local search algorithm described by Kauffman were run on stochastically generated NK fitness landscapes (100 iterations for each of 6 values of *K*). These results serve as a baseline for comparison with the modified algorithm described in the previous section. Additionally, 3 values (15, 30, and 45) for the "risk tolerance" were selected, and for each "risk tolerance" value 600 iterations were run (100 iterations for each of 6 values of *K*). Recall that a risk tolerance value of *x* indicates that *x* one-bit changes that yield a decline in fitness value will be tolerated before switching to a conventional local search.

First, results are presented for the average fitness values achieved for each of the 4 search algorithm variants as a function of *K*. In Figure 6, it can be seen that Kauffman's results are replicated, in that higher fitness values are achieved at relatively low values of *K*, and a "complexity catastrophe" occurs as the value of *K* approaches *N*. However, it is somewhat difficult (and misleading) to make a comparison of the algorithms based on the results in Figure 6 because as the value of K changes, the characteristics of the fitness landscape change. For example, a fitness value of 0.65 on an NK landscape where *K*=2 is not the same as a value of 0.65 on an NK landscape where *K*=8. This difference results from two facts that are often not noted in the management literature:

1) Holding N constant, the average fitness value decreases as *K* increases.
2) As *K* increases, the variance in fitness values decreases with the exception of a small number of local optima that are higher than those found for low values of *K*. In effect, as *K* increases, there become fewer clear "good solutions" while at the same time, the other local optima become more inferior.

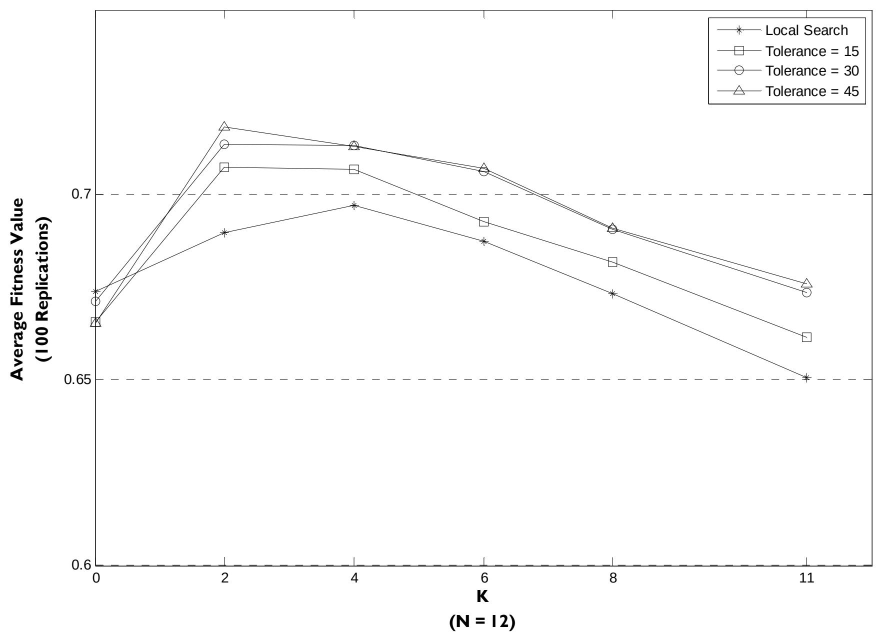

Figure 6. Comparison of search algorithm performance with respect to average maximum fitness achieved for varying risk tolerance values.

Therefore, Figure 7 plots a comparison of a metric that better illustrates the difference between search behaviors. It can be seen that at the extreme, where *K*=0, the global fitness value is always achieved as described by Kauffman. It can also be observed that at *K*=2 there is a significant difference between the local search algorithm and the search algorithm where a total of 15 changes resulting in "negative steps" are allowed. Another observation is that the two algorithms with higher tolerance for "negative steps" (e.g. values of 30 and 45) yield increased performance over the other two algorithms as complexity increases. Finally, it is noted that the algorithm that is most tolerant of "negative steps" (tolerance =45) escapes local optima, but does not consistently outperform the case where tolerance =30. This suggests there may be a limit or threshold to how much "learning" by trial-and-error may be beneficial in an NPD project.

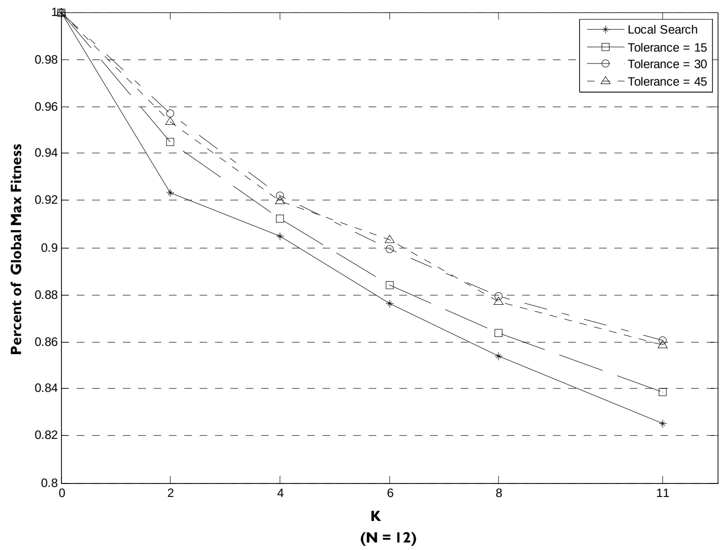

Figure 7. Comparison of search algorithm performance with respect to percentage of the global maximum fitness value achieved for varying risk tolerance values.

An investigation into the cost of increased system performance is shown in Figure 8. Here, it can be seen that in each search algorithm, the cost of search declines as *K* increases. This result is due to finding a local optimum faster for higher values of *K*, simply because there are more local optima as *K* increases.

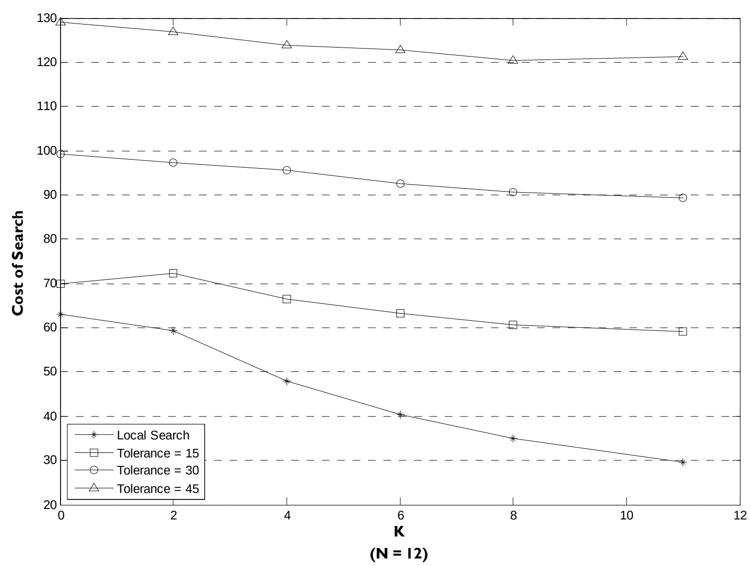

Figure 8. Comparison of Search Algorithm performance with respect to total cost of search for varying risk tolerance values.

It can also be observed that the risk tolerant search algorithms do not decrease in cost with *K* at the same rate as the basic local search algorithm. This result stems from the fact that the risk tolerant algorithms search the landscape for a given number of total negative steps before converting to a traditional local search, whereas the local search algorithm finds its solution in fewer steps because it does not explore for possible higher peaks on the landscape and, thus, conserves search cost. Finally, it can be observed that for the extreme where *K*=0, the increased cost of search provides no benefit because there are no local optima to escape and the conventional local search will always find the configuration that yields the global optimum fitness value.

3.5. Conclusion and Implications

In this chapter, our objective was to begin to examine how the NK model might be used to understand NPD as a complex set of decisions. To this end, we critically analyzed the NK model and some of its assumptions. Then, based on our understand of the NK model and NPD, we modified and extended the NK model to an NPD setting in which 1) firms do not necessarily reject moves that appear unpromising, but rather can explore the landscape to possibly escape becoming trapped on local optima and 2) there are costs associated with exploration and the reversal of decisions.

Viewing the results presented in the previous section in the context of product development, there are important implications and insights to be gained. First, by exploring the landscape characterized by multiple product design interdependencies, and recognizing not all "steps" will yield increased fitness, it becomes possible to escape local optima and eventually achieve higher fitness values. In a product development context, this suggests that firms can "learn" about the fitness landscape on which they find themselves by conducting experiments. Not all experiments will produce promising results, but pursuing some initially unpromising paths can result in finding better system designs in the long run. Further, the relative system design (fitness) is enhanced through learning and tolerating "negative steps" as the complexity of the product / system increases.

Second, several researchers (for instance, Levinthal, 1997; Sommer and Loch, 2004) have demonstrated that local optima can be escaped using a procedure known as distant search, in which multiple elements of the system are changed simultaneously. In the context of the NPD literature, this is often equated to radical innovation (as opposed to incremental innovation). However, distant search (sometimes called long jumps) has the characteristic of causal ambiguity in complex systems. In other words, when multiple elements of a complex system are changed simultaneously, it is difficult to determine which element or combination of elements led to the improvement in fitness. Thus, if future incremental changes are to be made, it can be difficult to determine which element(s) should be changed. On the other hand, in the search described in this paper, causal ambiguity is eliminated. So, while the search may take longer than a distant search, the search provides more knowledge about the landscape which, in the end, may prove to be more valuable for future product / system upgrades.

Regarding the costs associated with search for new product designs, the costs associated with search are abstracted in their meaning and are assumed to be normally distributed in this paper. Other distributions may be more appropriate (e.g. Triangular, Beta, Gamma), but an implication of the results in this paper regarding cost of search is that the cost-benefit tradeoff, in general, becomes more pronounced as product / system complexity increases. In other words, the increased costs of a more risk tolerant approach to product development may become more beneficial as the product complexity increases. This is especially true if we recognize that, as complexity increases, there become fewer winning (highly differentiated) solutions, so the impact of *not* finding a solution that is high relative to the global maximum can be especially deleterious. Finally, as observed in Figure 6 and Figure 7, there are limits to improvements resulting from increased experimentation. Therefore, firms should not assume that more comprehensive search (at increased cost) will yield better results. Instead, there is a threshold where firms gain almost no value from additional search and, thus, are better served by spending time and money finding the best local optima they can with the knowledge they have gained about the fitness landscape.

Our study also contributes to the theory of NPD modeling. Specifically, we find that the NK model, in its original form, is not an "off-the-shelf" model for NPD. For instance, the NK model does not model the cost of search activity—all decisions and moves are assumed to be free, which is obviously not the case. We have shown one possible way to model search activity costs in this study. Additionally, we have shown that the default agent search behavior in the NK model is not consistent with NPD activity, in that product development is *not* a monotonically increasing process. Instead, some exploration of initially less-fit alternatives is often beneficial and necessary, and we have explicitly modeled that aspect of NPD in this study.

One of the strengths of the NK model is its ability to produce generalizable results from a relatively small set of parameters. These results can provide valuable insights into the dynamics of a complex system such as a NPD project. However, the nature of the generalized results are, at the same time, one potential limitation of the NK model in that it is not entirely clear how to interpret and operationalize the results for specific applications. Thus, future research directions exist in exploring the development of complex products / systems using the NK model, namely:

Nature of Dependencies. In the NK model (and in this study), all dependencies are assumed to be of an unknown nature. That is, when a change is made to a given component, each of the other components which it impacts are assigned new fitness values from the uniform distribution. In NPD, there is generally some understanding of how components and decisions interact, at least on a pair-wise basis. We may not be able to accurately quantify the magnitude of the each dependency as there is always uncertainty associated with each dependency. However, modeling known complementary dependencies (mutually beneficial) and conflicting dependencies (trade-offs) would be beneficial to understanding complexity in NPD.

Uncertainty in Decisions. In addition to complete ignorance regarding dependencies, it is assumed that there is no knowledge regarding the outcome of design changes (mutations) in the NK model. In NPD, however, a probability distribution other than the uniform may be more appropriate for modeling the outcome of design changes, due to these changes being conscious, informed decisions rather than random mutations.

Multiple teams working on the same project. Whether it is multiple sub-teams within a single firm, or a multi-firm development effort, NPD is often a collaborative effort of multiple teams rather than one large development team. Thus, it would be beneficial to use the NK modeling framework as a starting point to link multiple efforts together in order to enhance our understanding of how coordination impacts NPD outcomes.

These research directions hold promise for further exploring how the NK model might be applied to an NPD setting and enhancing our knowledge regarding complexity in NPD. Therefore, the next two chapters of this dissertation are dedicated to rigorous study of these issues in order to more fully answer our initial research questions: "Is it possible to use complexity science to inform NPD?" and "Can the NK model be applied to the domain of NPD?"

Chapter 4: Extending the NK Model: Part 2

4.1. Introduction

As we showed in the previous chapter, the NK model holds potential as a means by which to study NPD, owing to its ability to produce emergent behaviors at the system level through interacting decisions. However, to this point we have also seen that the NK model is based on a few assumptions that do not necessarily align with NPD dynamics. In this study we further examine and extend the NK model to provide a more realistic model of the NPD process. We do this by incorporating two important aspects of NPD management in our model: 1) managers and engineers have knowledge of complementary versus conflicting dependencies[6] within an NPD project and 2) changes to components are subject to uncertain, but not completely random, results. The examination of these two NPD realities allows us to further the research on NPD using the NK model because previous research utilizing the NK model concentrated almost exclusively on how the search behavior of agents (Baumann & Siggelkow, 2012; Chao & Kavadias, 2008; Frenken, 2006; Lenox et al., 2007; Sommer & Loch, 2004) and the structure of dependencies (Ghemawat & Levinthal, 2008; Rivkin & Siggelkow, 2007) impacts performance of the organization. Further, the NK model has received some criticism in its application to organizations due to its somewhat

[6] Complementary dependencies describe a relationship between two components of a system in which an improvement to component A leads to an improvement in component B (but is not necessarily a bi-directional relationship), whereas a conflicting dependency has the opposite effect and can be thought of as a tradeoff. The use of the term complementary is adapted from the economic literature used to describe complementarities within economic systems (Milgrom & Roberts, 1990).

abstract nature. Here we provide a more realistic, yet parsimonious, model of NPD using the NK framework by modeling the nature of dependencies, and the uncertainty regarding component level changes within NPD projects.

This study produces three insights. First, we find our extended NK model and simulation results suggest the nature of dependencies between system elements can moderate the effect of system complexity: when a system has a low degree of complementary dependencies, system performance is relatively unaffected by complexity, but when that same system contains a moderate to high degree of complementary dependencies, system performance increases with increased complexity. This insight is counterintuitive in that it is widely believed that more dependencies in a system have a universally deleterious effect. Second, our study highlights that NPD development times may be longer than the original NK model suggests. When we model component changes using a skew triangular distribution, as opposed to the uniform distribution specified in the original NK model, we find development times are longer and system improvements are more incremental. Finally, we uncover the tension between development time and product quality that is inherent in NPD, whereas the original NK model suggests that, as complexity increases, development time decreases. Specifically, we find the level of complementary dependencies in a system not only moderates the effect of complexity on system performance, but also moderates how complexity impacts the trade-off between system development time and system performance: when complementarities are few, as complexity increases, system performance declines but system development time is reduced; however, when complementarities are many, both system performance and system development time increase as a function of increased complexity.

The remainder of this chapter is organized as follows. Section 4.2 further critically reviews the NK model and some of its mathematical constructs that are less applicable for modeling the NPD process. Section 4.3 describes our extended NK model and methodology. Section 4.4 presents the results of our simulation experiments. Finally, Section 4.5 discusses our findings and broader implications for NPD theory and practice as well as potential directions for future research.

4.2. NPD and the NK Model

In an attempt to find a competitive advantage, firms constantly engage in efforts to develop new and better product offerings. Firms undertake many types of activities in search for competitive advantage including R&D spending (Chen & Miller, 2007), capital investment (Greve, 2003a), and the search for innovation (Greve, 2003b). Central to these activities is new product development (NPD), which is essentially a search for winning products in the marketplace.

There is large body of literature that argues search is boundedly rational (Simon, 1969) and "local" in nature, rather than exhaustive (Greve, 1998). In NPD terms, this implies firms search for new products that are similar to those they already produce because they already possess the necessary resources, costs are generally lower for incremental improvements, there are fewer technical risks, and there is willingness to satisfice (Cyert & March, 1963; Simon, 1979).

Local, or incremental, change in Kauffman's NK model means that only one component is changed before the overall system fitness is evaluated. Additionally, in Kauffman's original NK model specification, if the system fitness improves as the result of a change to one component, then the organization adopts the new configuration and continues searching. However, if system fitness decreases, the organization chooses a new component to change. Search continues in this manner until no further improvement can be made via one-component changes. This search process is equivalent to an adaptive walk (Kauffman, 1993).

Adaptive walks generally end by finding a local optimum, thus there is risk that a firm may get trapped on a suboptimal local peak. As a consequence, it is sometimes necessary to search more distantly (e.g. changing more than one component at a time) in order to discover higher performing regions of the landscape (Levinthal, 1997). Distant search (sometimes called a "long jump") has the characteristic of causal ambiguity: when multiple elements of a complex system are changed simultaneously, it is difficult to determine which element or combination of elements led to the improvement in fitness. If future incremental changes are to be made, it can be difficult to determine which element(s) should be considered (Rivkin, 2000). For this reason, search agents are thought to be unable to effectively consider distant search alternatives (Siggelkow & Levinthal, 2005). But, it has been argued that appropriate organizational design may allow for more effective distant search (Knudsen & Levinthal, 2007). In this study, we restrict our analysis to the widely used local search strategy described by Kauffman (Kauffman, 1993).

The NK model stochastically generates a performance landscape which agents search for high-performing configurations of components. We remind ourselves that the original NK model was developed in the study of evolutionary biology to investigate the role of gene interactions and mutations in the evolution of genomes. Typically, we think of evolution in a biological sense, but the development of new products is also evolutionary, in that firms search, through trial and error, for solutions to a complex search problem defined by interactions among components. As discussed in the previous chapter, the NK model would appear well-suited for understanding how new products evolve and are governed by the factors of size (*N*) and complexity (*K*). However, we now discuss two additional assumptions which underpin the NK model that are not congruent with the contextual realities of NPD.

4.2.1. Complementary and Conflicting Dependencies

The first assumption of the NK model that does not align well with NPD is that when design changes are made to a selected component, which we refer to as the *focal component*,[7] the effect on dependent components is completely random. In NPD, there is generally some knowledge about how changes to one component will impact other components. For instance, managers and engineers typically know whether there will be a complementary and/or conflicting impact on other components as a result of a change to their own component. Therefore, we implement the following function to assign values to the dependent components as a function of the focal component change[8]:

$$y' = \begin{cases} y + \dfrac{(x'-x)}{(1-x)}(1-y) & \text{for complementary (positive) dependency, } x' > x \\ y + \dfrac{(x'-x)}{(1-x)}(-y) & \text{for conflicting (negative) dependency, } x' > x \\ y & \text{for complementary or conflicting dependency, } x' \leq x \end{cases} \tag{2}$$

In equation (2) y' is the new value of the affected component, y is the previous value of the affected component, x' is the new value of the focal component and x is the previous value of the focal component.

This method of assigning fitness values to the dependent components specifies that changes in the dependent components are proportional to the change in the focal component, and is closely aligned with the notion of structured dependence between components introduced by Solow et al. (1999). Proportionality, in their model, is defined based on the ratio of the actual change to the maximum possible change toward either extreme (zero or one). For example, if the focal component improves from a fitness value of 0.4 to 0.7, it has moved 50 percent of the distance toward the maximum value of 1 and, therefore, all the dependent components will also move half

[7] as a opposed to the dependent components which are affected by the change to the focal component

[8] It should be observed from equation (2) that when the focal component, x, decreases in fitness, the dependent components retain their previous values because we assume that unsuccessful local changes to components are deliberately not implemented in the system (or product).

the distance toward the maximum value of 1. Negative changes are handled in a similar fashion. Using this method of assigning fitness values to dependent components, Solow et al. (1999) showed that increases in the value of K do not lead to a *complexity catastrophe*. A *complexity catastrophe* is the effect of higher complexity (K) leading to increasingly rugged landscapes with local peaks that proliferate in number and are less differentiated from the overall landscape (Kauffman, 1993). However, Solow et al. (1999) only examined uniformly positive or negative influences; that is, if the focal component improves, all dependent components also improve. Likewise, if the focal component declines in fitness, each dependent component also declines in fitness value. In our model we allow for the reality found in NPD that some dependencies are complementary while others impose a conflicting constraint which is what often leads to rework and iterations in the NPD process.

4.2.2. Assignment of Component-Level Fitness Values

Another assumption of the original NK model that does not align well with evolutionary NPD processes is that design changes to a focal component result in a new fitness value drawn from the uniform distribution on the unit interval. This assumption implies the outcome of a design change is completely random. However, in the NPD context, this method of updating the focal component is not necessarily realistic because if the fitness value of a given component is greater than 0.5 it suggests there is a greater than 50 percent probability of the updated fitness value being *less than* the previous fitness value. In the NPD context, changes to components (often referred to as engineering change orders) do not always result in an improved design; however, if the probability of improvement were not at least 50 percent, few managers would be inclined to approve the change. We model this reality by introducing the following method of updating the fitness value, x, of focal component:

1. Assign a probability, p, that the modified component will result in an improved design.
2. Model the updated fitness value of the focal component, x', using a triangular distribution with parameters: *min*, *max*, *mode*, p, with the mode equal to the current value of the focal component, x, and the *max* equal to:

$$max = x + \frac{(1-x)}{2} \tag{2}$$

which says the maximum improvement resulting from a given change is half the distance from the current fitness value to the theoretical maximum of 1. From these two parameters (*mode* and *max*), along with the value of *p*, we can compute the *min* parameter according to:

$$min = \frac{mode - max * (1 - p)}{p} \tag{3}$$

By using the triangular distribution with the parameters just described for assigning fitness values to the changed focal component, we capture the fact that design changes made during the NPD process are subject to uncertainty, but are not entirely random as implied by the original NK model.

We now turn our attention to integrating the modified assumptions described in this section into an extended NK model for NPD which captures contextual nuances which have previously not been modeled using the NK model framework.

4.3. Research Design and Methods

To provide a more realistic model of the NPD process we incorporate two important aspects of NPD management in our model: 1) managers and engineers have some knowledge regarding the existence of complementary versus conflicting dependencies within an NPD project and 2) changes to components are subject to uncertain, but not completely random, results. Before adding these components to the model, we provide the reader with a description of the original NK model.

4.3.1. Model Description

4.3.1.1. Implementing the Original NK Model

Our approach begins by first coding Kauffman's NK model of fitness landscapes in the R statistical computing language. To generate the fitness landscapes, we follow the details and pseudo-code given by Altenberg (1997). To obtain a non-trivial sized fitness landscape we selected a system size of *N*=12 which yields $2^{12} = 4{,}096$ possible system configurations. In accordance with Kauffman's original model specification, interactions within the system are defined by a *local* (adjacent neighbor) pattern of interaction. For example, if *K*=2, each component in the system is affected by its two nearest neighbors. In other words, each component is dependent upon *K* other components in determining its fitness. A graphical depiction of the *local* interaction pattern is presented in Figure 9.

X	X										X
X	X	X									
	X	X	X								
		X	X	X							
			X	X	X						
				X	X	X					
					X	X	X				
						X	X	X			
							X	X	X		
								X	X	X	
									X	X	X
X										X	X

Figure 9. The "local" pattern of interaction used in this study. In this figure, N=12 and K=2.

The pattern in Figure 9 represents a specific type of interaction structure, but in practice a particular interaction pattern (either existing or proposed) could be used to study how changes to a given product architecture could influence system-level outcomes.

In this study, agents search for superior configurations via adaptive walks as described in Section 4.2. Additional search strategies can certainly be modeled in which agents search less locally by expanding the number of components they change at a given time (Gavetti & Levinthal, 2000; Rivkin & Siggelkow, 2007), or by dynamically adapting their search behavior (Battiti et al., 2008). However, in this exploratory study, we maintain the local search strategy which has been empirically shown to closely represent human problem solving behavior (Billinger et al., 2010).

4.3.1.2. Introducing Complementary and Conflicting Dependencies

In this modeling extension, we modify the manner in which the fitness values of dependent components are updated when a linked component is changed. We follow Solow et al.'s (1999) NK/D model of structured dependency with one exception—we omit the distance factor which they calculate as $|i - j|$, or the number of components separating component *i* from component *j*. We omit this factor because, in our study, we do not associate distance (physical or otherwise) with the strength of dependency between components *i* and *j*. We will refer to this model as *All Complementary NK (AllComp)*, for when a given component is improved, all of its dependent components are updated in a proportional fashion. In this study, we also create a *No Complementary NK* (*NoComp*) model to examine what happens when improvements to a focal component result in changes to dependent components that are all inversely proportional.

4.3.1.3. NPD Extensions

Finally, to add contextual realism to the NK model for the study of NPD, we incorporate the two (2) new modeling constructs described in Section 4.3 of this study.

Mixture of Complementary and Conflicting Dependencies. We first extend the NK model by recognizing that interactions between component pairs in NPD projects are neither completely unknown (as in the original NK model), nor are the interactions all mutually reinforcing (Solow et al., 1999) or all characterized as design conflicts. Rather, in an NPD project some dependencies between components are complementary, while others are characterized by conflicts or tradeoffs. Therefore, we explicitly model different mixtures of complementary and conflicting interactions. In this study we use three treatment levels: *Low Complementarity (LC), Moderate Complementarity (MC), and* High *Complementarity (HC)* in which 25%, 50% and 75%, respectively, of the dependencies between components are complementary.

Focal Component Fitness. We then modify the manner by which a component selected for change (a focal component) is assigned a new fitness value. As discussed, changes to components during the NPD process are conscious design decisions and are not completely random in their result; however, there still remains a stochastic aspect to the success of a change. Therefore, we employ a triangular distribution to model the fact that drastic improvement (nor drastic decline) is highly probable, though still possible. We also model a probability of success (likelihood that the changed component will yield an improvement), which shapes the triangular distribution. For this study, we nominally set the probability of success, p, equal to 0.75. In terms of the triangular distribution, this means that 75 percent of the triangle's area lies to the "right" of the mode. (For simplicity, p is assumed to be time stationary and apply globally to each component in the system).

Figure 10 presents a graphical summary of our incremental modeling approach.

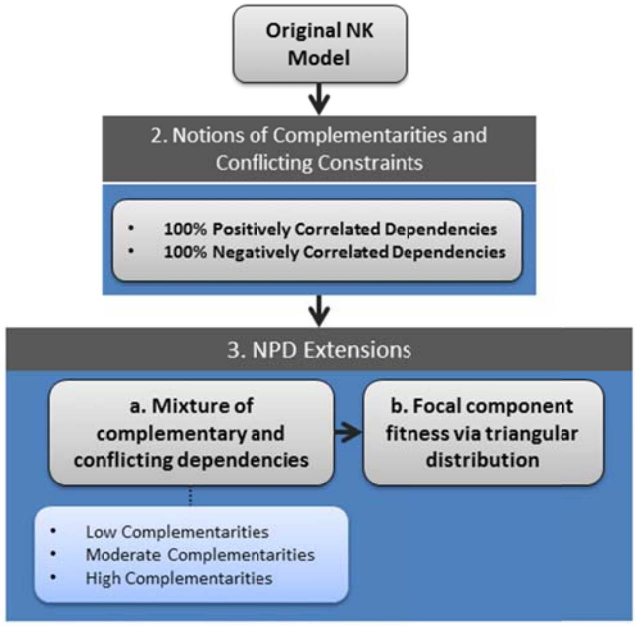

Figure 10. Methodology for modeling NPD as a complex system of interacting design decisions. Using the NK model as a starting point, modifications are made to introduce and explore alternative modeling constructs resulting in an NK model extended to the NPD domain.

4.3.2. Model Investigations

In this study we are interested in how our extensions to the NK model compare with its original specification. The measures of performance we are interested in are: 1) the maximum overall system fitness F achieved and 2) the number of candidate changes considered before stopping as a result of reaching a local optimum at which no further one-component changes can improve the system fitness. These two metrics directly relate to NPD measures of product quality and development time[9]

To control for variance due to differences in initial conditions (a hallmark of complex system behavior), each experimental run is conducted with the same set of initial starting configurations. Table 2 highlights the specific modifications and parameter settings for our modeling approach.

[9] Additionally, cost is acknowledged to be an important measure in the NPD context, but it is not explicitly modeled in this study as costs are very specific to each industry and individual project. However, in practice, costs associated with components and change orders (if available) could be modeled via probability distributions, similar to how fitness values are modeled in this study, either in scaled or absolute terms using appropriate probability distributions.

		Structured Dependencies		Mix Complementary and Conflicting Constraints			Focal Component modeling via Triangular Dist		
	A. Original NK Model	B. AllComp NK	C. NoComp NK	D. Low Complementarity	E. Moderate Complementarity	F. High Complementarity	G. Low Complementarity	H. Moderate Complementarity	I. High Complementarity
Focal Component Fitness	Uniform (0,1)	Uniform (0,1)	Uniform (0,1)	Uniform (0,1)	Uniform (0,1)	Uniform (0,1)	Triangular	Triangular	Triangular
Depdent Component Fitness	Uniform (0,1)	Proportional to Focal	Inversely Proportional to Focal	Mixture of Proportional / Inverse Proportional	Mixture of Proportional / Inverse Proportional	Mixture of Proportional / Inverse Proportional	Mixture of Proportional / Inverse Proportional	Mixture of Proportional / Inverse Proportional	Mixture of Proportional / Inverse Proportional
Complementarities	N/A	100%	0%	25%	50%	75%	25%	50%	75%

Table 2. Comparison of models and parameters used in this study. Changes from the original NK model are highlighted as the model is incrementally modified (from left to right).

4.3.3. Model Application

To conclude our investigation, we use design information regarding a brake system from the research literature (Black et al., 1990) to illustrate the application of our models to an actual system. We use the specific pattern of dependencies for the brake system (

Figure 11), but we also collect information regarding the change history associated with each of the models. This allows us to examine the rate of convergence for each of the model specifications we examine. Finally, we test for significance between each of the models in a pair-wise fashion using Tukey's Honestly Significantly Difference (HSD) post hoc test.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1. Customer Requirements	X												
2. Sys Level Parameters	X	X											
3. Wheel Torque		X	X										
4. Piston Size- Front		X	X	X	X								
5. Piston Size - Rear		X	X	X	X								
6. Pedal Mechanical Adv.	X	X		X	X	X			X	X			
7. Lining Coeff. - Rear		X		X	X	X	X		X	X			
8. Lining Coeff. - Front		X		X	X	X		X	X	X			
9. Booster React. Ratio		X	X	X	X	X	X	X	X	X			
10. Rotor Diameter	X	X	X	X	X	X	X	X	X	X			
11. Booster - Max Stroke									X		X		
12. Caliper Compliance		X		X		X			X			X	
13. ABS Modulator Displ.												X	X

Figure 11. Interaction of components found in an automobile brake system. Source (Black et al., 1990)

4.4. Results

Having described the theoretical foundations and methodological framework for extending the original NK model to the NPD domain, we now describe the results of our simulations. In our study we are primarily interested in two measures of NPD performance: product quality (taken as the average of component fitness values) and development time (measured in terms of the number of simulation steps, where each step represents a consideration of a component design change). Product quality is important for obvious reasons, while the development time metric is important because managers of NPD teams and projects have finite development time horizons and faster development times can lead to "first-mover advantage" (Lieberman & Montgomery, 1988).

In each of our experiments we initialize our simulation (t=0) by randomly defining a product, $\omega = (x_1, x_2, \dots x_N)$, comprised of 12 components (N=12). During each time period (t = 1,2,3...), a change to one of the N components is made, resulting in a new product configuration, ω'. If $F(\omega') > F(\omega)$, then the new product configuration is adopted. This process of change, selection, and adoption is repeated in each time period until no further improvement to the product can be made by changing a single component.

4.4.1. Baseline Results—Original NK Model

Table 3 and Figure 12 illustrate how system performance varies with complexity (K) under the original NK model (Kauffman, 1993). Those familiar with the NK model will recognize the familiar inverted "U-shaped" curve, indicating the average maximum system fitness is greatest when the value of K is moderately low (K=3, in this case). This is well-known result in the literature. It provides a baseline for the remainder of our results.

Average Peak System Fitness

	A. Original NK	B. All Complementarities	C. No Complementarities
K=0	0.6624	0.6624	0.6624
K=1	0.6917	0.7409	0.6061
K=2	0.7016	0.7824	0.5225
K=3	0.7085	0.8236	0.5068
K=4	0.708	0.8456	0.4987
K=5	0.6987	0.8636	0.498

Note: Each result in the table above is an average of 200 landscapes. To control for variance due to differing starting configurations between simulations, the same set of 200 starting configurations was used for each result.

Table 3. Comparison of results obtained for average maximum fitness obtained for Original NK, *AllComp*, and *NoComp* models.

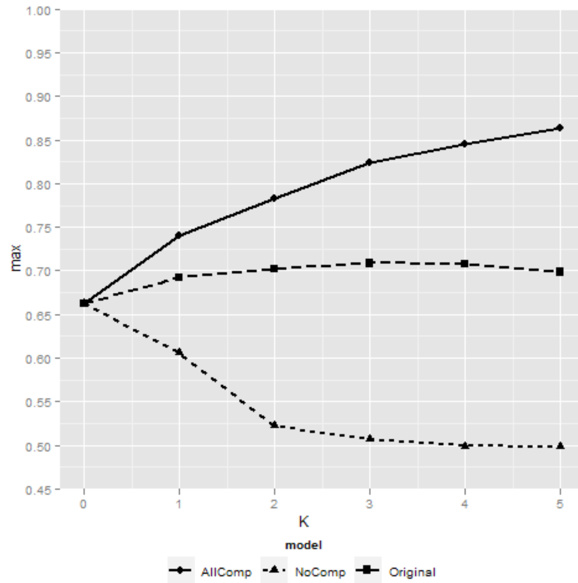

Figure 12. Average maximum fitness for Original NK, *AllComp*, and *NoComp* models. Each point represents the average of 200 simulation runs using the same set of initial starting configurations.

Table 4 and Figure 13 present a result of the original NK model not discussed in the extant literature concerning the time (simulation steps) required to reach a local or global optimum. As Figure 13 shows, a general pattern of shorter development time is associated with increasing complexity in the original NK model. We note, here, that this result does not align with intuition regarding development times in NPD: more complex systems typically take *longer* to develop. However, when we consider the search strategy described by an adaptive walk (local search), the result, though not necessarily realistic in an NPD context, makes sense because higher complexity in the NK model generates more "rugged" landscapes which are characterized by more local maxima, leading to shorter "walks".

Average Steps to Solution

	A. Original NK	B. All Complementarities	C. No Complementarities
K=0	50.48	50.48	50.48
K=1	47.51	48.49	46.85
K=2	42.395	43.085	24.67
K=3	38.395	37.88	16.825
K=4	35.61	34.345	13.89
K=5	32.355	32.435	13.04

Note: Each result in the table above is an average of 200 landscapes. To control for variance due to differing starting configurations between simulations, the same set of 200 starting configurations was used for each result.

Table 4. Comparison of results obtained for average number of design changes considered (simulation steps) for Original NK, *AllComp*, and *NoComp* models.

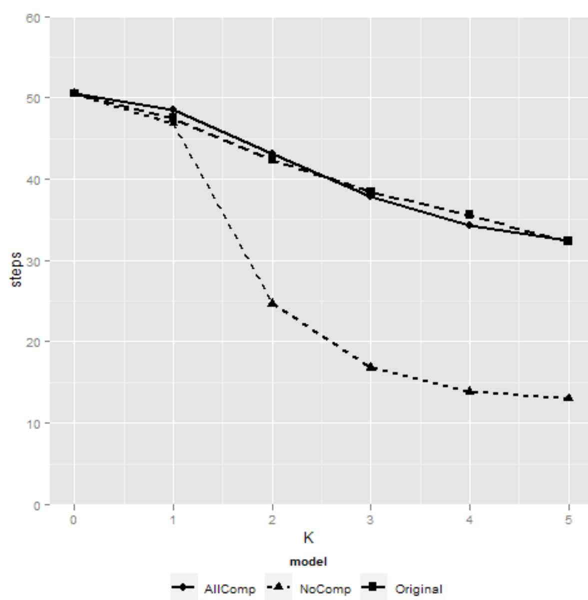

Figure 13. Development time (steps) for Original NK, *AllComp*, and *NoComp* models. Each point represents the average of 200 simulation runs using the same set of initial starting configurations.

4.4.2. Extreme Results—All Complementary & No Complementary Dependencies

We now examine the results of two extreme extensions of the NK model, namely the models in which all changes to dependent components are either positively or negatively proportional to changes in the focal component.

Table 3 and Figure 12 (above) illustrate the effect of these extreme extensions on system performance (fitness). In the model where all dependencies are positively correlated with the focal component, we observe that the performance of the system increases, in a logarithmic fashion, with respect to complexity (*K*) (Figure 12). However, when all dependencies are negatively correlated with improvements to the focal component, each interaction is a conflicting constraint and we observe that fitness values decline as complexity increases (Figure 12). In fact, as K increases, the maximum fitness values regress to the mean.

In terms of system development time, we again observe there is a pattern of shorter development times as complexity increases (Figure 13). However, it is worth noting here that the underlying reasons for the shorter development times as complexity increases in these two extreme NK model extensions are somewhat different than in the original NK model. In the original NK model, development times (steps) decrease in K as a result of increased ruggedness in the landscape. However, in the positively correlated model, development time decreases as a function of *K* because, when all dependencies are complementary, the rate at which system-wide improvements are made is increased drastically, resulting in faster convergence to a solution. On the other hand, in the model with no complementary dependencies, each conflicting dependency places an additional constraint on opportunities for system-wide fitness improvement and, therefore, the search for better product configurations terminates faster. This result does not necessarily follow intuition regarding complexity in NPD, and we will have more to say regarding this in the next chapter.

4.4.3. NPD Extensions

Having briefly examined the original NK model, as well as the two extreme cases of complementary dependencies, we examine the results of two extensions to the original NK model grounded in the contextual realities of NPD.

4.4.3.1. Mixture of Complementary and Conflicting Dependencies

The results from our experiments using the original NK model and the models which have uniformly positive or negative correlated dependencies are helpful in understanding that complementarities (and a lack thereof) have an influence on the fitness of a system as it evolves. However, in the context of NPD, dependencies between components of a system are almost always a mixture of complementarities and conflicts. We now examine what happens when mixtures of complementary and conflicting dependencies are present in an NPD project.

Table 5 and Figure 14 (solid lines) show how system fitness is influenced by the presence of both complementary and conflicting dependencies. When complementary (positively correlated) dependencies are low (25% in this study), system fitness declines as *K* increases, but at a slower rate than when there are no complementarities. It is interesting to note that when complementary dependencies are low, system fitness (for all values of $K > 0$) is less than when K=0. This was not an expected result, for it was believed that even a low number of complementary dependencies would be better than a system in which all components are independent. However, it is not until complementary dependencies account for at least 50 percent of the total number of dependencies that we see a benefit to additional complexity. Namely, when we increased the level of complementary dependencies to a moderate level (50%), we found that the number of complementary dependencies was sufficient to yield an increase in system fitness when compared to the case where K=0 (see Figure 14, Panel B, solid line). However, as K increases, the degree to which system fitness is improved effectively "plateaus". This occurs because, as K increases, the expected number of components that will benefit from an improvement in one component becomes nearly equal to the expected number that will experience a decline in fitness thus creating an upper limit on the average maximum system fitness. Finally, when we model a high level of complementary dependencies (75%) in the system (Figure 14, Panel C, solid line), we observe that system fitness increases quite rapidly with complexity. This result follows intuition that when complementary dependencies outnumber conflicting dependencies, system fitness increases with the number of total dependencies because complementarities can be exploited to a higher degree as K increases.

	Uniform	Triangular	t-test
Panel 1. Low Complementarity			
K=0	0.6624	0.5617	p=2.2e-16
K=1	0.6559	0.5615	p=2.2e-16
K=2	0.6073	0.5576	p=3.14e-12
K=3	0.5891	0.5531	p=3.27e-7
K=4	0.5738	0.5444	p=1.78e-5
K=5	0.5643	0.5399	p=0.00047
Panel 2. Moderate Complementarity			
K=0	0.6624	0.5617	p=2.2e-16
K=1	0.6885	0.5883	p=2.2e-16
K=2	0.6811	0.6152	p=2.2e-16
K=3	0.6895	0.6327	p=2.2e-16
K=4	0.6853	0.6479	p=8.45e-9
K=5	0.6894	0.6596	p=5.20e-6
Panel 3. High Complementarity			
K=0	0.6624	0.5617	p=2.2e-16
K=1	0.7131	0.6169	p=2.2e-16
K=2	0.7443	0.6746	p=2.2e-16
K=3	0.7653	0.7302	p=1.49e-7
K=4	0.7761	0.7852	0.1384
K=5	0.7873	0.8114	p=3.09e-5

Table 5. Comparison of results obtained for average maximum fitness using different methods of updating the focal component fitness value. All results are for the local pattern of interaction.

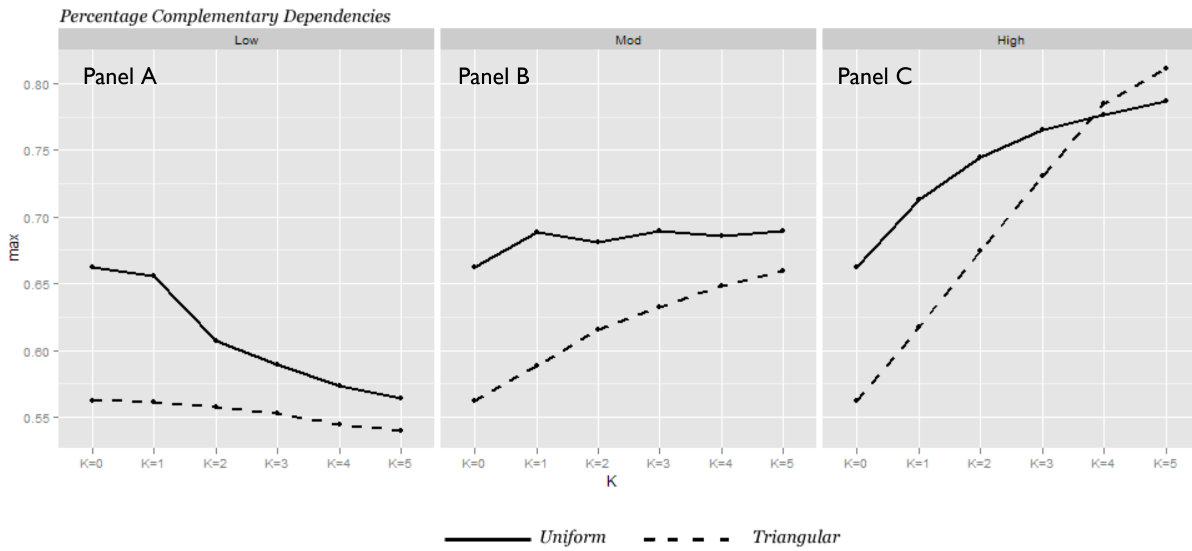

Figure 14. Mean maximum fitness for varying degrees of complementary dependencies and different methods of assigning new fitness values to the focal component. Each point represents the average of 200 simulation runs using the same set of initial starting configurations.

With regard to development time for an NPD project, when we combine complementary and conflicting dependencies into the same model in varying percentages (models D, E, and F in Table 2) we obtain results that, again, share the same general trend—decreasing development time as a function of *K*. The most notable decrease, as a function of K, is found in the model with low complementarities. This more pronounced decrease, in the low complementarity model, results from the fact that when few complementary dependencies exist, an increased number of total dependencies creates a situation in which improvements to a given component is almost always outweighed by the conflicts with other components, which leads to the search becoming frustrated very quickly. Table 6 and Figure 15 (solid lines) illustrate this phenomenon.

	Uniform	Triangular	t-test
Panel 1. Low Complementarity			
K=0	50.48	71.8	p=2.2e-16
K=1	43.15	54.83	p=2.11e-9
K=2	34.81	47.94	p=5.37e-14
K=3	31.07	44.12	p=2.27e-10
K=4	26.62	36.89	p=2.14e-8
K=5	25.26	32.42	pe=5.10e-6
Panel 2. Moderate Complementarity			
K=0	50.48	71.8	p=2.2e-16
K=1	45.18	64.89	p=2.2e-16
K=2	41.65	70.12	p=2.2e-16
K=3	40.7	72.83	p=2.2e-16
K=4	35.73	72.67	p=2.2e-16
K=5	34.65	69.25	p=2.2e-16
Panel 3. High Complementarity			
K=0	50.48	71.8	p=2.2e-16
K=1	46.62	77.08	p=2.2e-16
K=2	45.47	92.02	p=2.2e-16
K=3	41.59	104.87	p=2.2e-16
K=4	38.73	112.47	p=2.2e-16
K=5	36.13	108.22	p=2.2e-16

Table 6. Comparison of results obtained for average number of design changes considered (simulation steps) using different methods of updating the focal component fitness value. All results are for the local pattern of interaction.

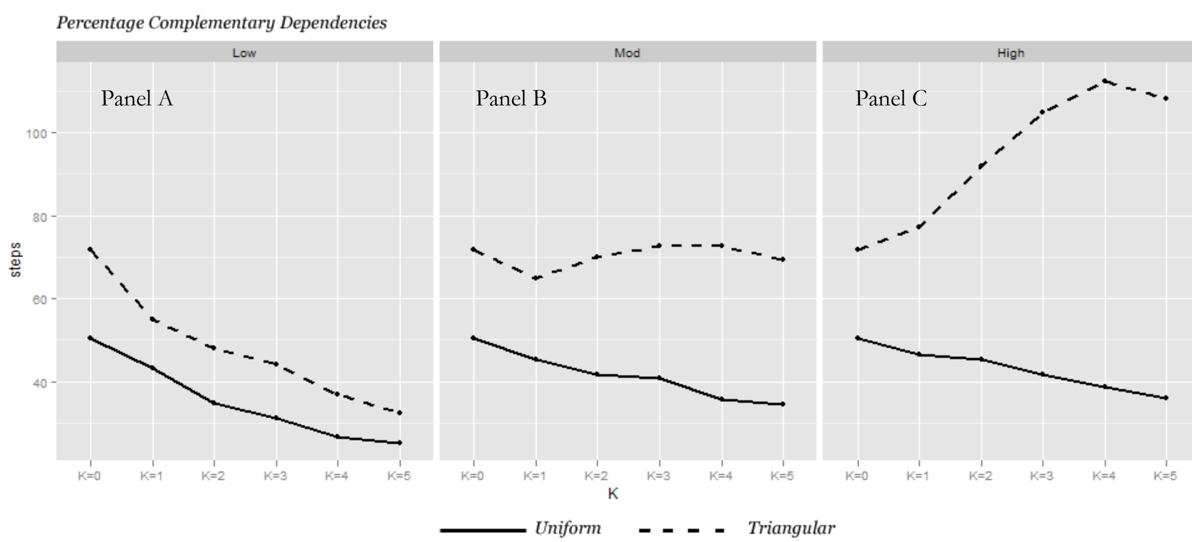

Figure 15. Mean development time (steps) for varying degrees of complementary dependencies and different methods of assigning new fitness values to the focal component. Each point represents the average of 200 simulation runs using the same set of initial starting configurations.

4.4.3.2. Focal Component Fitness Changes

Modeling changes to the focal component using the triangular distribution discussed in Section 4.3.2 allows us to account for the fact that the design changes are stochastic but not completely random in their result, and that improvements to components tend to be incremental rather than drastic in nature. The results of our extension to the original NK model using the triangular distribution for new focal component fitness values when a change is made (models G, H, and I in Table 2) are shown in Table 5 and Figure 14 (dashed lines). We first observe that, compared to the models in which fitness values for the focal component are distributed uniformly, system fitness decreases more slowly in *K* when complementarities are low (25%) and increases more rapidly when complementarities are moderate and high (50% and 75%, respectively). In other words, the slope of the line which plots fitness as a function of *K* is greater, in all cases, when changes to the focal component are distributed according to the triangular distribution. In our study, the probability of improvement, *p*, equals 0.75 which gives a higher probability of a component change resulting in improved fitness when compared to the original NK model in which the probability of a component change resulting in improvement is equal to: $1 -$ *current fitness of component*. We could debate whether p=0.75 is an appropriate parameter setting, but anything less than 0.5 would likely be too risky for most NPD managers. Ultimately, the result of this increased probability of component improvement is that there are more opportunities for component changes to result in component and system wide fitness improvement. Therefore, when complementary dependencies are low, system fitness still regresses as K increases, but at a slower rate. And, when complementary dependencies are equal to or greater than 50%, the increased probability of component changes resulting in system improvement yields a greater marginal benefit for each increase in *K*.

However, in Table 5 and Figure 14 (dashed lines) we observe that, in using the triangular distribution to model focal component changes, system fitness is, in almost all cases, less than when using the uniform distribution to model component changes in the original NK model. To understand why this happens, we note that in the triangular distribution we have chosen, the maximum improvement is limited to half the difference between a component's current fitness and the maximum value of 1. And, due to the peaked shape of the triangular distribution, component improvements are more likely to be incremental because new component fitness values are more likely to be close to mode of the distribution which, in our study, is the current component fitness value.

We now consider how using the triangular distribution in our extended NK model impacts development time. When we changed the distribution of new fitness values for focal component changes, we obtained results more aligned with what we would expect regarding the development time of complex systems. As shown in Table 6 and Figure 15 (dashed lines), development time is longer in comparison to the original NK model specification (solid lines) when we model component changes as being 1) more likely to succeed (p=0.75) and 2) more incremental by using the triangular distribution. This is because increased the probability of change success, coupled with incremental gains, leads to more possible paths for exploration as well as slower convergence to a solution, resulting in longer and more gradual adaptive walks.

Additionally, in Figure 15, Panel C (dashed line) we note when complementary dependencies account for a high percentage (75% in this study) of the total dependencies in the system, the length of the project development time increases with complexity. This occurs because, when there is a sufficiently high percentage of complementary dependencies in a system, adding more dependencies (increasing K) creates a situation in which additional dependencies increase the likelihood of improved system fitness when a change is made to a component. This, in turn, lengthens the overall search process because new paths for improvement are more likely to continue to be found. It is worth explaining, though, the result in Figure 15, Panel C (dashed line) where there appears to be a critical point at which the search time begins to decrease as a function of increased complexity (K=5, in this case). At first, we suspected this result may simply be an artifact of the stochastic nature of the simulation. However, repeated runs of the simulation using different random number streams yielded the same phenomenon each time. This result is explained as a result of two competing dynamics—the rate of system improvement and the likelihood of improvement. Specifically, we found high levels of complementary dependencies, coupled with a high probability of component improvement, create a situation in which the likelihood of system improvement increases in K which lengthens the overall search process. However, as K increases beyond a certain point, the system fitness improvement at each step becomes so large that a locally optimal solution is found faster. Thus, in Figure 15, Panel C (dashed line), when K=5, the search converges faster than when K=4, because the rate of system fitness improvement becomes so rapid that the number of steps required to find a local maximum for system fitness begins to decline beyond K=4.

4.4.4. Application Example

Table 7 reports the results obtained for our example brake system (Black et al., 1990), when we simulate its development using the models described in this study. This example system is obviously not a *local* pattern of interaction, but we note that that the system does have parameters N=13 and complexity equivalent to K=3.8 (computed by dividing the total number of off-diagonal dependencies by N). From Table 7, we first observe there are significant differences between almost each pair-wise comparison of models, both in terms of performance (fitness) and development time (steps). And, while the model with a low percentage of complementary dependencies does not differ significantly from the no complementarities model, in terms system performance, there is a significant different difference in terms of development time between these models, which leads us to our next observation.

	Max Fitness	*Steps*
A. Original NK Model	0.6798	39.87
B. Positive Correlation in Dependent Components	0.8197	42.21
C. Negative Correlation in Dependent Components	0.5325	26.75
D. Low Complementarity	0.5425	55.21
E. Moderate Complementarity	0.6266	84.88
F. High Complementarity	0.7295	129.4
Tukey HSD Result	All significant differences at α=0.05, except C-D	All significant differences at α=0.05, except A-B

Table 7. Performance (fitness) and development time (steps) for example brake system.

A salient result from Table 7 is that our NK model extensions, which capture two important NPD contextual elements (models D-I in Table 2), suggest development times are much longer than what the original NK model would suggest, given the identical search strategy and pattern of interaction. This is illustrated graphically in Figure 16 in which system development is plotted temporally with each line representing the mean fitness at each step for 100 simulations with identical initial configurations.

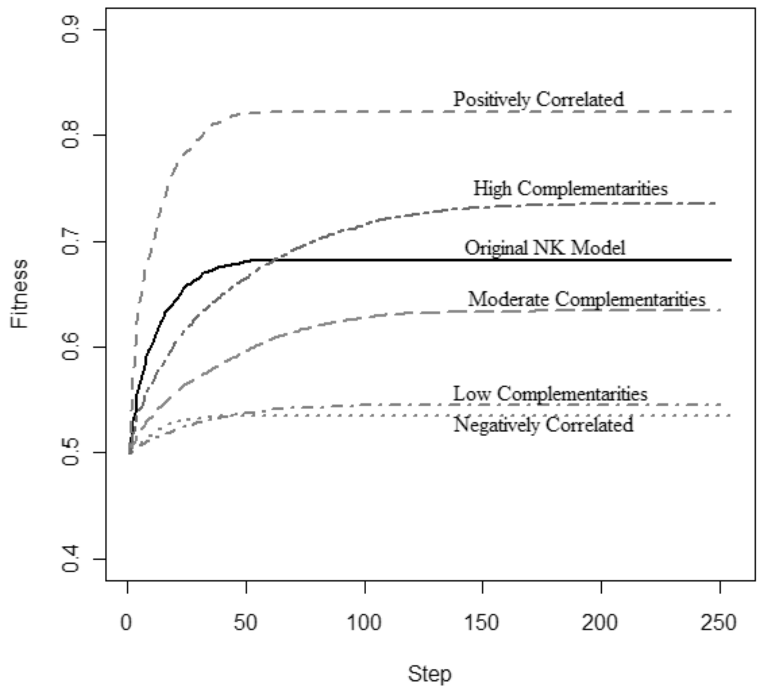

Figure 16. Temporal plot of fitness for example brake system as development evolves. Each line represents the average of 100 simulation runs using the same set of initial starting configurations.

In this example system, we do not have any information regarding the dependencies within the system (e.g. whether a given dependency is complementary or conflicting). Therefore, in each iteration of the simulation we characterize the system according to the 2 general modeling constructs introduced in this paper—percentage of complementarities and a new probability distribution for component fitness values when a component is changed. With the implementation of these two constructs we find informative results regarding two measures of performance. Specifically, if the system contains more complementary dependencies, system performance increases, but this comes with a trade-off of longer development times. Cost-benefit analyses could easily be conducted, but such an analysis would only be valid for the specific set of parameter values used in this study. However, more importantly, if more specific data regarding dependencies were available, we could obtain even more informative results regarding the design of a system of interest. The ability of the model developed in this study to inform NPD theory and inform the development of real world systems (with data elicited from subject matter experts) is an important contribution that will be discussed in the next section.

4.5. Discussion and Implications

4.5.1. Theoretical Implications

A number of previously published studies have used the NK model to investigate and gain insight into the interactions among multiple decisions within an organization. Due to small sample sizes, costs, and time, insights at the theoretical level regarding the outcomes of strategic new product development decisions are difficult to develop and test in the real world. However, with regard to the relationship between product complexity, product performance, and development time, our extensions to the NK model offer insights for NPD that are fundamentally different than what the original NK model would suggest, which opens new avenues for further research and model development.

A central result of our study is that the nature of interactions—whether they are complementary or not—within an NPD project matter as much or more than sheer number of interactions. Figure 14 (dashed lines) and Figure 15 (dashed lines) show that, in our extended NK model of NPD, if we hold K constant, the system performance and development times vary widely, depending on the nature of the interactions within the system. In other words, the degree to which dependencies are complementary versus conflicting moderates the effect of complexity on system performance and development time. Therefore, with this insight, we may be able to design interactions and interaction structures that can be exploited for more successful NPD. More specifically, our model highlights the importance of complementarities and also the importance of avoiding conflicting dependencies within a product. For instance, in Figure 14 we see that when there is a low level of complementary dependencies, system performance (fitness) monotonically decreases as complexity increases, suggesting that conflicting dependencies should be avoided, especially as product complexity increases.

Another theoretical insight from our study concerns the development time of an NPD project. In Figure 15, we observe that our extensions to the NK model (and the parameter settings) suggest the development time of NPD projects is consistently longer than under the original NK model. This result is also illustrated in our example brake system (Figure 16) in which we note that system improvements are more incremental than the rapid initial gains suggested by the original NK model. We argue that this dynamic of more incremental performance improvement more accurately models how NPD projects typically evolve.

Our study also reveals another relationship that has an important theoretical implication that differs from the original NK model of complexity. Namely, Figure 14 (dashed lines) and Figure 15 (dashed lines) show that, in our extended NK model of NPD, performance and development time are often negatively correlated. That is, as system performance increases (favorable), development time also increases (unfavorable). This result is in contrast to the original NK model which suggests that there is a moderately low level of complexity (*K*) at which system performance is higher than at other values. It has not received much attention in the extant literature, but at this critical value of *K* the adaptive walk length is shorter than at lower values of *K*, suggesting that we are able to have dual benefits—faster development *and* better system performance. However, our model calls attention to the fact there is a constant tension and trade-off between product quality and development time.

4.5.2. Implications for Practice

Because our extensions to the NK model capture additional contextual realities of the NPD process, an important implication for practice is that the extensions described in this study can be a more realistic model for NPD when compared to the original NK model. At a practical level, the modeling constructs and framework presented in this paper offer a relatively easy tool for exploring various strategic product architectures via Monte Carlo simulation. For instance, the removal or addition of a specific component interaction can be explicitly modeled via the interaction matrix which is familiar to practitioners due the popularity and prevalence of the DSM tool (Steward, 1981). Additionally, sensitivity to uncertainty in the probability of design change success can be modeled using the extensions described in this study in conjunction with Monte Carlo simulation techniques. Finally, by virtue of being a computational model, the assumptions and parameters in the model can be easily modified in order to test specific hypotheses regarding candidate product design alternatives.

4.5.3. Limitations and Future Research

The model we have presented is exploratory in nature. We have described and illustrated at least two ways in which the NK model can be adapted to more realistically simulate and explore the complex process of NPD. However, we recognize this model has limitations which create opportunities for further research in this domain. First, this model remains relatively abstract in nature. For example, the metric of fitness, measured on a scale of 0 to 1, is a proxy for system performance, but such a metric is rather abstract. Therefore, further research in the applied domain could investigate how to map component performance and system performance to more meaningful metrics via methods such as conjoint analysis (Green & Srinivasan, 1978). Another limitation of our study is that we did not implement more sophisticated search behaviors in our model. However, our intent was not to find optimal search strategies. Our objective was to develop and explore a new modeling methodology for NPD based on the popular NK model. Testing hypotheses regarding alternative search behaviors to the adaptive walk described in this paper could prove to be a fruitful avenue for future research.

Additionally, our model assumes a time homogeneous fitness landscape. In other words, the nature of the interactions and the probability for improvement at each step of the simulation does not change as the system evolves. Further research on shifting fitness landscapes presents an opportunity to explore how periodic—and often exogenous—changes to the interaction structure of a product impact the development process. Another extension to this study could involve further characterizing the nature of dependency interactions within a system by modeling dependency *strengths*. In other words, rather than characterize each dependency as complementary or conflicting (positive or negative effect), future research could explicitly model the fact that some dependencies are stronger than others by scaling dependencies on the interval (-1,1). Finally, in addition to the finding that the nature of interactions within a system are important, exploration of the how interactions are configured could be an important research opportunity, combining the concepts in this study with the notion of patterned interactions introduced by Rivkin and Siggelkow (2007).

4.6. Conclusion

In summary, the NK model has gained widespread acceptance in the management science and organizational design literatures. This study critically analyzed the NK model and several of the assumptions that underpin this popular model of complexity, and found that the NK model neatly captures one of the central problems found in NPD organizations—dependency between components—but could be improved for modeling NPD by addressing two aspects of NPD that are not congruent with the NK model: knowledge about dependencies within a system, and design changes that have inherent uncertainty, but are not completely random in their outcomes. We then proposed and implemented two extensions to the NK model that address these two nuances regarding NPD as it relates to the NK model. We call attention to how the simple, yet powerful, NK model can be adapted and extended to model the complex process of NPD while avoiding over-parameterization in the model. A central result of our study is that the nature of dependencies within a system is perhaps more important that the raw number dependencies or interactions in a system. However, perhaps the most important aspect of this study is the development of a modeling framework that can be used to gain further insight into the general domain of product development. Finally, our extended NK model for NPD can support the development of specific system-level hypotheses about the outcomes of real world projects which can then be tested with real world parameters regarding component-level dependencies and component-level change probabilities.

Chapter 5: Beyond the NK Model: Co-evolution in NPD

5.1. Introduction

The NK model has been widely applied to organizational search in the context of competitive strategy (e.g. Levinthal, 1997). For instance, Ganco and Hoetker's (2009) review of the NK model and its use in the management literature found over 30 publications in leading management and strategy journals that utilize the NK model. And, in the preceding chapters, we have examined how Kauffman's NK model of stylized fitness landscapes, developed in the field of evolutionary biology, can be extended in several ways to better capture the realities and dynamics of NPD. However, despite the NK model's widespread use, the NK model assumes all decisions are evaluated at the system level and does not account for a fundamental aspect of many NPD projects: that projects are subdivided into sub-teams whose decisions are often based on improving their own subsystem, rather than in the interest of improving the entire NPD project.

This research utilizes Kaufmann's (1993) variant of the popular NK model, called the NKC model, to explore how an ecology of design teams interact within an NPD environment characterized by complexity. The goals of this study are to answer key questions regarding interacting teams and inform engineering and NPD managerial practice. Our research is further motivated by the following questions:

1. Given an overall project size, how does the partitioning of the project into work teams of different sizes affect project outcomes of quality and development time? In other words, what partitioning strategy leads to better outcomes: a project with fewer larger sized teams or

a project with a greater number of smaller sized teams? Answering this question holds potential value in determining the appropriate, or even optimal, team size for a given level of complexity.

2. How does an increased number of interacting decisions *between* teams affect project quality and development time? Reducing interactions between subsystems is commonly known as product modularity. We seek to better understand how modularity impacts key NPD measures of performance, namely quality and schedule.

To explore these research questions, we model NPD design activities and decisions using Kauffman's NKC model of co-evolutionary species and make three contributions. First, our results reveal that the number of interacting decisions within an NPD project is not necessarily a good indicator of project complexity. Second, we show that the structure of design interactions, rather than the number of interactions, is more important in determining project outcomes of interest. Finally, our results suggest that there is a limit to the benefits of product modularity. Furthermore, this study finds that the NKC model is more appropriate for modeling NPD activity as compared to the basic NK model which has previously gained widespread acceptance.

The remainder of this chapter is organized as follows. Section 5.2 contains a discussion of related research. Then, Section 5.3 presents our model for investigating how interacting teams making local decisions affect system level outcomes. Our data and results are described and analyzed in Section 5.4. Finally, Section 5.5 concludes with a discussion of our results and their implications for practice, give qualifications on our findings, and provide concluding remarks.

5.2. Related Literature

In this section we discuss how this study relates to three areas of related research: 1) Organizational Complexity, 2) Product Modularity and 3) Search on a Fitness Landscape.

5.2.1. Complexity in Organizations

In the organizational literature, there are two salient lines of research regarding complexity in organizations. In the first line of research, scholars have suggested that the number of organizational parts give rise to complexity in organizations (Daft, 2009; Jablin, 1987). The number of parts, it is argued, matters because there is an implicit assumption that further subdivision and differentiation of tasks requires additional integration of the organizational parts and their outputs.

In another line of research, it is argued that complexity in organizations is related to the interactions among the parts of the organization. Foundational work in self-organization (Nicolis & Prigogine, 1977) led others to develop models to explore the emergent behavior of systems comprised of many interacting particles and genes (Fontana, 1990; Kauffman, 1993). While these models of interacting particles and genes have their roots in the fields of chemistry and biology, the insights and methodologies have been applied and had a clear impact on organizational studies (Anderson, 1999; Marion, 1999; Rivkin, 2000; Siggelkow & Levinthal, 2005; Sommer & Loch, 2004; Baumann and Siggelkow, 2012).

In this study, we explore how these two strands of research on organizational complexity are related by investigating the following question: "Is there a difference in the performance of similarly sized NPD projects based on the number of interacting sub-teams in the project?" For similarly sized projects, fewer sub-teams imply sub-teams that are larger in size. We are interested in how the sizing and number of teams in a project influence the performance and development time of projects.

5.2.2. Product Modularity

A line of research related to organizational complexity concerns the design and management of new products using modular architectures. Not only are organizations complex, by Simon's (1962) definition, but the products they design and manufacture are often characterized by complexity. In other words, many products involve a large number of parts that interact in non-trivial ways. In Simon's (1962) pioneering essay on complex systems, he argued that nearly decomposable systems help reduce the complexity associated with system design. Nearly decomposable systems have come to be synonymous with modular product architectures because such architectures group interdependencies within units called modules while minimizing the interdependencies among modules. A system with higher modularity, then, implies that more of the interdependencies in the system are contained within the individual modules.

Several research efforts have argued that product modularity is a means by which firms can manage complexity (Langlois, 2002; Parnas, 1972; Ulrich & Eppinger, 1999). Other benefits of modularity in product design have been argued as well such as better organizational coordination (Sanchez & Mahoney, 1997), mass customization (Baldwin & Clark, 2000) and codification of design standards which promotes easier outsourcing (Schilling & Steensma, 2001). In this study, we explore:

1. How product modularity affects the performance and development time of projects.
2. How project size and interdependencies within a project impact its eventual performance and development time.

5.2.3. Searching on Fitness Landscapes

In developing new products, firms continually experiment with new technologies and opportunities in the search for better product offerings in the marketplace. Firms engage in NPD experiments with many possible combinations of technologies, engineering designs, manufacturing processes, organizational forms, etc. This experimentation can be characterized as a *search* for winning configurations of elements under the firm's control. Confounding the firm's search, however, is the fact that, often, some of these elements (technologies, designs, processes, etc.) are interdependent. That is, a change to one element can have an effect on other elements.

As previously discussed in this dissertation, Kauffman (1993), borrowing from Sewell Wright's notion of *fitness landscapes* (1932), published the NK model which uses fitness landscapes to characterize a search. Configurations of elements that are favorable have a higher elevation (fitness) on the landscape, while less favorable combinations are associated with valleys, or lower fitness, in the landscape. The terrain of a landscape is determined by the degree of interdependence between elements. When there are many interdependencies between elements, the result is a rugged landscape marked by many peaks of differing elevation. Conversely, few or no interdependencies between elements generate a smooth landscape in which there are few peaks and each location on the landscape is closely correlated with nearby locations.

Formally, as discussed previously, the NK model is defined as follows. A system consists of N components, and each component makes a contribution to the fitness of the overall system via the fitness function:

$$F = \frac{1}{N}\sum_{c=1}^{N} f_c$$

Each of the N components can be in one of *A* states, but many researchers (including this study) reduce the dimensionality of the search space by setting *A* equal to two, without loss of generality. Thus, for a system comprised of *N* components, the number of possible configurations of the system is 2^N. A second parameter *K* defines the degree of interdependency between the components of a system. Specifically, *K* represents the number of other components in the system that affects the fitness of each component. Thus, the value of K can range from 0 to N-1 and each component can assume one of 2^{K+1} fitness values. In Kauffman's model, fitness values are taken to be uniformly distributed random variates on the interval (0,1).

Search on a fitness landscape is modeled as an adaptive walk (Kauffman & Levin, 1987) in which one or more elements of a given configuration are modified. If the change results in a configuration with a higher fitness, the new configuration is adopted and search continues from the new location on the landscape. On the other hand, if system fitness decreases as a result of the new configuration, the system retains its previous location and chooses a new element to change. Search continues in this manner until a local optimum is found—a point on the landscape at which all neighboring points have a lower elevation or fitness. Occasionally, the stopping point will be highest point in the entire landscape in which it is also the global optimum.

Despite its roots in the biological sciences, the NK model has been an attractive analogy for investigating organizations, especially in the context of interdependent decision making (e.g. McKelvey, 1999; Levinthal and Warglien, 1999). However, for all its popularity in modeling organizations and interdependence, the NK model contains an underlying assumption that is not congruent with how many NPD projects are organized. Namely, it is assumed that changes to the configuration of a system are made and evaluated centrally. In the context of NPD, this would imply that decisions regarding component level changes are made by executives, and subsequent evaluation of the changes could be made centrally as well. However, as previously discussed in this section, responsibility for decisions in NPD projects is often decomposed into modular teams across an organizational hierarchy because the technical and market expertise required for effective product development is possessed by engineers and development personnel, not executives. Additionally, due to the size and scope of many NPD projects, responsibility is often assigned to design teams according to expertise and product functionality (e.g. in the developing a new aircraft there may a dedicated team responsible for the fuselage, one for the flight controls, another for the wing assembly, etc.).

5.3. *Model Framework*

5.3.1. A Co-evolutionary Model of NPD Dynamics

While the NK model neatly captures the essence of complex systems and emergent behavior which results from interactions between constituent parts, it does not account for multiple groups making local decisions, but interacting with and affecting the performance of other groups. Therefore, to model the NPD search process, we acknowledge that multiple teams are responsible for the design of subsystems which themselves interact and integrate to form a complete product or system. Each team makes decisions based on improving their own subsystem, but such decisions often have impacts on other subsystems within the project. Thus, we adopt Kauffman and Johnsen's (1991) NKC model of co-evolutionary fitness landscapes. In applying the NKC model to NPD, we operationalize the following parameters:

S—the number of co-evolutionary design teams involved in the NPD project

N—the size of each co-evolutionary design team, e.g. the number of elements for which each team is responsible. This is the same *N* found in Kauffman's original NK model. For simplicity, each team is assumed to be equal in size in this study.

K—a measure of interdependency *internal* to each design team. The fitness of each of the *N* decisions for a given design team is influenced by *K* other decisions internal to that team.

C—the first of two variables that determines the interdependency *external* to each design team. *C* is the *number of other design teams that influence* the fitness of each element within a given design team.

Y—the second of two variables that determines the interdependency *external* to each design team. *Y* is the *number of elements within the C design teams* that influence the fitness of each element within a given design team. Therefore, the product of Y and C represents the density of external dependencies for each element in a given design team.

To illustrate the difference between how the NK and NKC models represent and model an identical system we examine Figure 17.

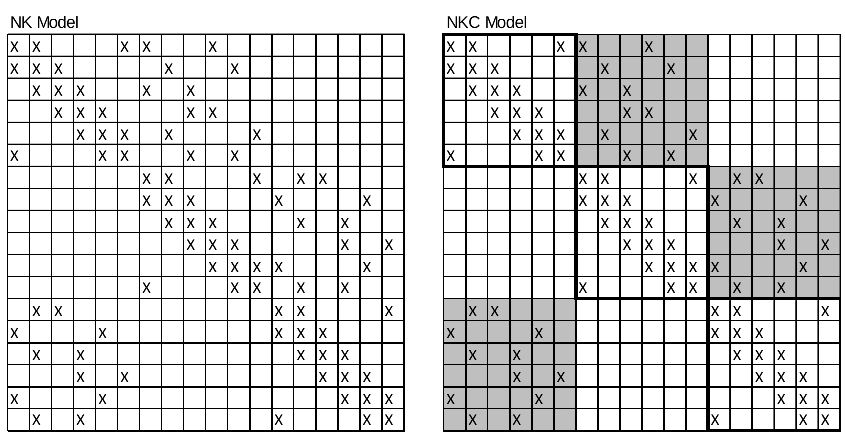

Figure 17. Comparison of identical project dependency structures as represented in the NK model and NKC model.

In the NK model (left), each "X" represents a design dependency, similar to the design structure matrix (DSM) representation (Steward, 1981). In this example project, the system size is represented by the parameter N=18, and the complexity parameter K=4, indicating that each component's fitness is influenced by itself (main diagonal "X"s) plus the state of 4 other components. Thus, the row sums of the NK interaction matrix are all equal to K+1=5.

In Figure 17, the NKC model (right) has the same placement of the "X"s in the system. However, the NKC model representation shows that there are three subsystems (S=3), each of which have a size of $N' = 6$ and an internal interaction complexity of $K' = 2$ (outlined matrices). These subsystems, in the NKC model representation, are influenced not only by other components in the subsystem itself, but also by components in other subsystems (shaded matrices). In this example, each component is influenced by 2 other components (C=2) from 1 other subsystem (Y=1)[10].

The primary difference between the NK and the NKC models relates to the behavior of the search. In the NK model, a random component is selected for modification and if the change results in an overall fitness improvement to the system, the change is adopted and the search continues until no one-component changes lead to a fitness improvement for the system. In the NKC model, however, changes are evaluated only at the subsystem, or team, level. Each team, in turn, makes a change and then evaluates its impact on its *own* subsystem performance. If the change results in higher subsystem fitness, then the change—and the resulting new configuration—is adopted, regardless of the impact on the rest of the system. Having highlighted this important difference we now describe, in more detail, our implementation of the NKC model as applied to NPD.

At the beginning of each simulation (t=0), the project is initialized with S teams. Each of the S teams is responsible for a subsystem, $\omega = (x_1, x_2, \dots x_{N'})$, comprised of N' components. The fitness of each subsystem is initialized by assigning to each of the N' components a random variate from the uniform distribution on the unit interval. Thus, the fitness of each team's subsystem $F_{subsystem}$ is given by:

[10] To avoid confusion in notation between the NK and NKC models, the terms N* and K* are introduced used when describing the size and dependency parameters for the each sub team in the NKC model. Thus, it can be seen that N from the NK model is equivalent to $N' * S$ from the NKC model $(18 = 6 * 3)$; and K from the NK model is equivalent to $K' + (C * Y)$ from the NKC model $(4 = 2 + (2 * 1))$.

$$F_{subsystem} = \frac{1}{N'}\sum_{x=1}^{N'} f_x$$

Then, the fitness of the overall project $F_{project}$ is defined by the average fitness of all the teams' subsystems:

$$F_{project} = \frac{1}{S}\sum_{\omega=1}^{S} F_{subystem}$$

At each step of the simulation (*t=1,2,3*), one of the *S* teams modifies one of its *N* components, which yields a new subsystem configuration, ω', in which the changed component and *K* other components in the subsystem have new fitness values. If $F(\omega') > F(\omega)$, then the new subsystem configuration is adopted by the team; otherwise, the previous configuration is retained. If a new subsystem configuration has dependencies in other subsystems, the fitness values of the dependent components are also updated with a new random variate. As a result, the fitness landscapes of the project teams are coupled because the decision by one team to modify a component can affect other teams within the project. (For higher the values of *C* and *Y* in the simulation, it is more likely that a component in one subsystem will be impacted due a change in another subsystem.)

For subsequent steps in the simulation, the *S* teams continue to make changes, in turn, with each team trying to improve its *own* subsystem fitness by modifying one of its components. Some of the component modifications will improve a team's subsystem fitness and also simultaneously improve other subsystems' fitness via design dependencies. However, it will often be the case that an improvement to one subsystem will cause the fitness of other subsystems to decline. This process of co-evolutionary change, selection, and adoption is repeated in each time period until a) a Nash equilibrium is reached, indicating that no further improvements to subsystem fitness can be made by any team by changing a single component or b) the number of simulation steps reaches a predetermined upper limit, realizing that some systems will never reach a Nash equilibrium.

In this study we are interested in how team size, the number of teams, and the degree of interdependency in a project—both within teams and between teams—impact 1) the overall project fitness and 2) the time to reach a Nash equilibrium, or the point at which each team is at a local optimum and cannot improve its own fitness via a one-component change. These two metrics directly relate to NPD project performance measures of product quality and development time. Therefore, in each simulation run, we collect and record the overall project fitness value at each step of the simulation as well as the number of simulation steps required before a Nash equilibrium is reached[11].

Though the NK model has been far more popular than the NKC model in the management and strategy literature, the NKC model has been used in this domain. Notably, the NKC model was used as the framework for Levitan et al. (2002) to study optimal team size in organization. However, our study differs in two important ways: 1) the size of projects in their study was held constant at 100 total agents and 2) each team (group) in their study was assumed to be fully connected to all agents in the same group. In our study, we relax both of these assumptions as we explore different project sizes as well as varying levels of interdependency within (and between) groups.

5.3.2. Model Verification

For this study, we wrote a simulation model of Kauffman's NKC model of co-evolutionary dynamics in the R statistical computing language which is available from the author upon request. However, because little research has been conducted using the more computationally complex NKC model (as compared to Kauffman's NK model) we first endeavored to verify that our simulation model behaved in the manner that Kauffman (1993) described. Figure 18 illustrates an example of our simulation results of the temporal co-evolution of a large system in which there are 6 groups (S=6), each composed of 24 elements (N=24). Each of the groups has an internal coupling that is relatively high (K=13); in other words, the fitness of each of the 24 elements in each group is influenced by 13 other elements in the group. Additionally, each element's fitness is not only impacted by other elements within its own group, but also by 1 element (C=1) in each of the other 5 groups (Y=5).

[11] As in real-world NPD, not every simulated project will reach a Nash equilibrium. Therefore, if a project does not converge to a Nash equilibrium, we report the max steps parameter and report project fitness as the value achieved at t(max steps).

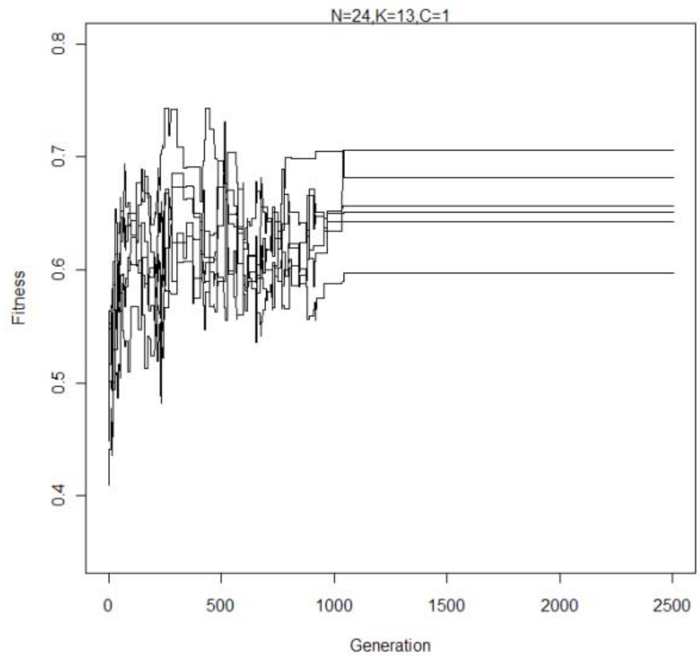

Figure 18. Temporal evolution of the fitness of 6 co-evolving teams using the NKC model. In this particular, a Nash equilibrium is reached, but not all co-evolutionary sets of teams will reach Nash equilibrium, even for identical parameters. This is due to the stochastic nature of the fitness landscapes and initial conditions.

Figure 18 shows that in this system the groups do, in fact, co-evolve to a Nash equilibrium after approximately 1100 simulation steps. It can also be observed that the groups—and the system as a whole—initially experience a period of rather rapid increase in their fitness. This initial rise in fitness then levels off and enters a period of "churn" in which each group continues to search for a better configuration and, when successful, causes changes to the other groups' fitness. The groups then continually adapt to changes made by the other groups until each group has found a configuration that is more fit than all possible one-component variations. Our results in Figure 18 replicate the dynamics of the original NKC model (Kauffman, 1993, p. 246, Figure 6.2). Our model verification efforts also found that, as noted earlier, and as found by Kauffman (1993), it is entirely possible for a set of co-evolutionary groups to not reach a Nash equilibrium, depending on the initial conditions of the groups and the coupled fitness landscapes. It can also be seen in Figure 18 that the co-evolutionary dynamics of the NKC model produce a "messy" process, not unlike that found in NPD, in which each group struggles to continually adapt to changes brought about by changes in other parts of the system. Finally, Figure 18 also reveals that the eventual configuration found by each group may not be optimal for the subsystem, but is the best that can be achieved, given the actions of the other groups, which also reflects a dynamic found in NPD activities. Confident that our simulation model and results accurately replicates the original NKC model specification, we now model NPD using the NKC model to investigate our research questions by performing a series of 4 experiments as outlined in Table 8.

Experiment	Purpose	Method
1	Compare results of the original NK model and the NKC model	Model projects with identical interdependency structures and initial configurations using the NK model and NKC model
2	Explore the relationship between team size and the number of teams	Partition equally sized projects by varying the parameters S (number of teams) and N (size of each team), while holding other parameters constant. This allows comparison of projects that have the same overall size and total number of interdependencies within the project.
3	Examine how intra-team and inter-team design dependencies impact project outcomes	Vary all parameters in the NKC model to create project interdependency structures that vary in modularity. Modularity in this experiment is defined as the ratio of interdependencies for each component that are internal to the team versus external to the team.
4	Investigate the efficacy of an alternative decision heuristic for teams	Prohibit sub-teams from adopting changes that do not yield at least 5 percent improvement to their subsystem design.

Table 8. Outline of experiments performed in Chapter 5.

5.4. *Results and Analysis*

5.4.1. Experiment 1: NK vs. NKC Results

If the NK and/or NKC models are to be used in NPD research, it is important to first understand how the NK and NKC models compare in their representation of NPD projects. Our first experiment, therefore, is designed to systematically examine the how the NK and NKC models perform, given the same interaction patterns within an NPD project. Each of the projects in this this experiment has an overall size of 60 components (N=60 for the NK model, while we use $S = 4$ and $N' = 15$ for the NKC model to generate projects of equivalent size). We vary the number of interdependencies in each of the projects by adjusting the parameters C and Y, while holding the K' parameter constant at K'=4. To control for variance due to differences in initial conditions (a hallmark of complex system behavior), each model is given the same set of initial starting configurations—the same initial fitness values and interaction patterns.

Maximum Fitness

Figure 19 and Table 9 present the simulation results for overall system fitness under the NK and NKC model specifications outlined above. Each point in Figure 19 represents the average of 200 independent runs of the simulated evolution of an NPD project. Additionally, it should be noted that the parameter on the x-axis refers to the number of design dependencies per component. In the NK model this is the *K* parameter and in the NKC model this is equal to $K' + (C * Y)$ which is equal to *K* from the NK model. (The numbers in parentheses on the x-axis indicate the number of intra-team dependencies and inter-team dependencies per component, respectively).

From Figure 19 we make observations regarding the average maximum system fitness obtained using local search. First, we note that the average maximum system fitness is noticeably different under the two models, despite having the same initial conditions *and* number of dependencies per component. Specifically, we note that the maximum system fitness achieved under the NK does not vary significantly with team size or the complexity parameter *K*, though, as expected, the average maximum fitness does decrease slightly as additional dependencies are added. This result stems from the fact that the NK model represents the NPD project as one large team, rather than multiple interacting sub-teams. By representing the NPD project as one centrally managed team that evaluates each component change at the global system level, the NK model only considers two parameters, *N* and *K*, and does not distinguish between interactions within sub-teams and between sub-teams. This suggests that the structure of the interdependencies is less important in the NK model than in the NKC model which does, in fact, represent the NPD project as a group of teams that act locally, but have influence on other teams.[12]

[12] In this experiment all project interdependency structures have values of K', C, and Y that are uniform for all components in the project. Other interdependency structures have been investigated (e.g. Rivkin and Siggelkow, 2007) in which components vary in the degree to which they influence other components or are influenced by other components. Our concern, in this particular experiment, is how project outcomes compare using different two models, given the same overall interdependency pattern.

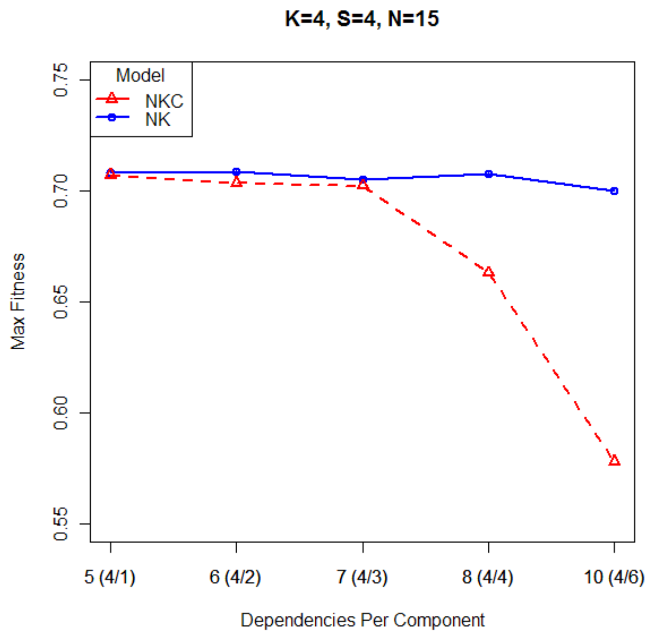

Figure 19. Comparison of results for overall average maximum project fitness achieved using identical project design dependency structures under the NK model and NKC models.

On the other hand, the results in Figure 19 reveal that under the NKC model, projects with identical dependency structures experience a much more pronounced decline in the average maximum fitness achieved as the number of dependencies per component increase. In this experiment, the additional dependencies are in the form of additional "external" dependencies. Thus, in the NKC model it can be seen that, as subsystems and components becomes more influenced by components in other subsystems, the overall maximum fitness of the overall system declines much more dramatically than the original NK model would suggest.

NK Model					NKC Model						
	Dependencies			Avg Max		Dependencies					Avg Max
Project Size	*Total*	*K*	*Intra/Inter*	Fitness	Project Size	*Total*	*K'*	*C*	*Y*	*Intra/Inter*	Fitness
N=60	300	5	NA	0.7081	S=4, N'=15	300	4	1	1	4/1	0.7072
N=60	360	6	NA	0.7087	S=4, N'=15	360	4	2	1	4/2	0.7036
N=60	420	7	NA	0.7051	S=4, N'=15	420	4	3	1	4/3	0.7022
N=60	480	8	NA	0.7076	S=4, N'=15	480	4	2	2	4/4	0.663
N=60	600	10	NA	0.6999	S=4, N'=15	600	4	2	3	4/6	0.587

Table 9. Comparison of numerical results for overall average maximum project fitness achieved using identical project design dependency structures under the NK model and NKC models. For each row in the table, the overall project size and the total number of design dependencies are equivalent.

Second, we note that there is a sharp decline in the average maximum fitness achieved once the number of external dependencies per component equals the number of internal dependencies per component. In this experiment, this "phase transition" occurs in the NKC model when $K' = 4$

and $C = 2; Y = 2$ (total number of dependencies per component equals 8). The same decline in system performance is not reflected in the original NK model because the NK model does not distinguish between internal and external subsystem dependencies or the local sub-team behaviors regarding decisions to adopt local improvements.

In sum, one central result of this experiment is the observation that, in contrast to the NK model, the average maximum system fitness found under the NKC model varies to a greater extent as we vary the number of dependencies within a given project. Specifically, the NKC model is capable of reflecting that additional dependencies external to each subsystem tend to impede the search for better solutions, whereas the NK model does not reveal this phenomenon. This result follows from the notion of product modularity. In this case, when we increase either *C* or *Y*, and hold *K* constant, we effectively create a situation in which product modularity decreases because we are reducing the "density" of interdependencies *within* each subsystem relative to the "density" of interdependencies between subsystems. Our results thus reveal the NKC model is able to capture and reveal one of the benefits attributed to modular product design—concentrating interdependencies within modules or sub-teams helps manage complexity and leads to better system performance.

Nash Equilibrium (Development Time)

Similar to our discussion regarding maximum system fitness, Figure 20 and Table 10 present a comparison between the NK and NKC models, in terms of the average number of steps to reach Nash equilibrium for identical design dependency structures. We first observe the time to reach Nash equilibrium is qualitatively and quantitatively much different in the NK model, when compared to the NKC model. Specifically, we note that in the NKC model, when the ratio of internal subsystem dependencies to external subsystem dependencies is relatively high, the average time to reach Nash equilibrium is faster (fewer steps) than what the NK model would suggest for an identical dependency structure. It can also be observed that in the NKC model, when the ratio of internal subsystem dependencies to external subsystem dependencies is relatively low (that is, when the number of external dependencies outnumber the internal dependencies), the average time to reach Nash equilibrium is significantly slower. In fact, in many cases, the process does not converge to equilibrium before the 2,500 step limit is reached. This result is due to the fact that co-evolutionary dynamics in the NKC model cause improvements in one sub-team to often knock other teams off their previous improvement location on their own fitness landscape. This dynamic is not unlike that found in actual NPD organizations in which one design change generates rework and additional design iterations for other teams.

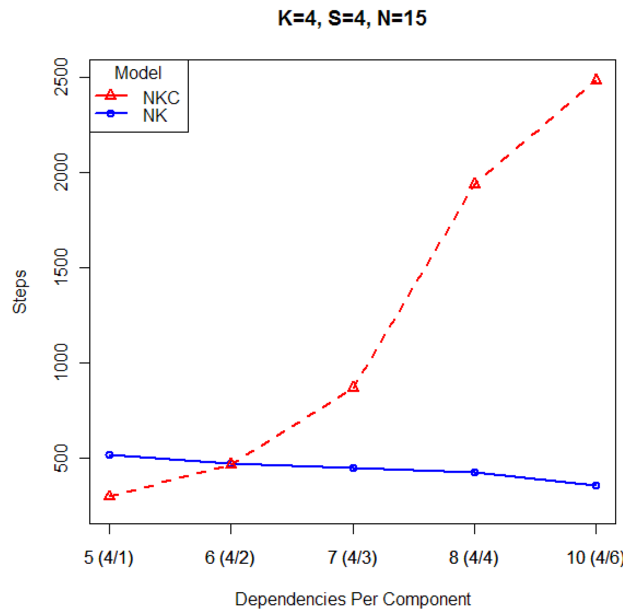

Figure 20. Comparison of results for average time to reach Nash equilibrium using identical project design dependency structures under the NK model and NKC models. Each point represents the average of 200 independent simulation runs.

NK Model					NKC Model						
	Dependencies			Time to Nash		Dependencies					Time to Nash
Project Size	*Total*	*K*	*Intra/Inter*	Equilibrium	Project Size	*Total*	*K'*	*C*	*Y*	*Intra/Inter*	Equilibrium
N=60	300	5	NA	520.71	S=4, N'=15	300	4	1	1	4/1	299.4
N=60	360	6	NA	474.85	S=4, N'=15	360	4	2	1	4/2	467.1
N=60	420	7	NA	449.77	S=4, N'=15	420	4	3	1	4/3	869.2
N=60	480	8	NA	425.16	S=4, N'=15	480	4	2	2	4/4	1937
N=60	600	10	NA	357.8	S=4, N'=15	600	4	2	3	4/6	2483

Table 10. Comparison of numerical results for average time to reach Nash equilibrium using identical project design dependency structures under the NK model and NKC models. For each row in the table, the overall project size and the total number of design dependencies are equivalent.

From our simulation results, we further observe that the effect of additional dependencies can have opposite effects in the NK and NKC models. For instance, under the NK model, additional dependencies effectively reduce the time to reach Nash equilibrium. This result is consistent with previous research using the NK model in which additional dependencies generate increasingly "rugged" fitness landscapes which lead to shorter adaptive walks because there are more local maxima on which to become trapped. In contrast, in this experiment, as additional dependencies were added, the effect was that the time to reach Nash equilibrium was dramatically *increased*. The idea that additional dependencies leads to longer searches which correspond to NPD development time is consistent with intuition. This result follows from the fact that the additional dependencies in this experiment were all between subsystems as opposed to within subsystems. This finding leads us to note that, under the NKC model, dense coupling of design decisions within the sub-teams, relative to the entire system, leads to shorter development times, whereas many inter-team dependencies leads to substantially longer development times. However, the same dynamic is not captured by the NK model because it does not explicitly model the effect of several linked NK models; instead it treats the dependency structure as one large, centrally managed team. The results of further investigation into the relationship between internal and external sub-team dependencies, and its effect on project performance (fitness) and development time is presented later in this section.

5.4.2. Experiment 2: Team Size vs. Number of Teams

Having examined the difference between the simulation results for the NK and NKC models, we now focus on exploring a central motivating question of this paper, which is, whether and how does team size and the number of teams on an NPD project affect the project's outcomes. We retain the NKC model for this experiment in favor of the NK model due to its demonstrated ability to model the co-evolutionary nature of design teams within an NPD project. We set up 5 stylized NPD projects with varying number of total interdependencies within each project. For each of the stylized projects we hold the overall project size and total number of interdependencies constant, and examine the effect of dividing the project into 4 differently sized project sub-teams. We, therefore, simulate 20 different project structures (5 levels of total interdependencies and 4 methods of partitioning each project). For each of the 20 different project structures, we ran 200 independent simulations. Additionally, for run *i* of 200 in each experiment, we use pseudo-random number stream *i* to control for variance due to initial conditions. We report the results of our experiment, first presenting the outcomes in terms of project quality (maximum fitness), and then describing the development time (Nash equilibrium) results.

Maximum Fitness

Figure 21 illustrates the average maximum fitness achieved for the simulated NPD projects under 4 different project partitioning schemes. First, observing the performance comparison between projects with a smaller number of larger-sized teams and projects with a greater number of smaller-sized teams we find, contrary to our initial hypothesis, few significant differences in performance (see Table 11) using Tukey's honestly significant difference (HSD) test to correct for multiple comparisons. One notable significant difference in average maximum fitness is that the projects with 10 sub-teams underperform the other projects with fewer sub-teams when there are relatively few dependencies between components from different sub-teams. This result is explained by the fact that, despite the fewer number of inter-team dependencies, the greater number of teams adds a higher "cost of coordination". This coordination cost manifests itself in a higher probability that, in any given round of the simulation, another sub-team will make a design change that will negatively impact another sub-team by virtue of having more sub-teams.

Interestingly, however, the projects with a smaller number of larger-sized teams (e.g. S=4, N'=15) begin to underperform the projects with a greater number of smaller-sized teams when the number of dependencies between components from different sub-teams is relatively high. This inverse effect can be seen in Figure 21 where three of the lines intersect the line representing the S=10, N'=6 partitioning scheme. We can understand why the projects with fewer sub-teams underperform as the number dependencies increase in this experiment if we consider the source of the increased inter-team dependencies where this observation occurs. Specifically, we note that as the value of Y (the number of other sub-teams with which each sub-team interacts) increases (from 1 to 2 to 3), it becomes particularly high relative to the number of sub-teams, S. The result, then, is that in each round of the simulation, the projects with a high value of Y relative to S, there is an increased probability that changes in one sub-team will negatively impact another team. Conversely, in the projects where the value of Y, relative to S, is lower (e.g. Y=3 and S=10), there is a lower probability that changes made in one sub-team will negatively impact another team. Thus, we find another possible strategy in managing the cost of coordination in projects: at the strategic level, it is advantageous to minimize not only the number of total dependencies between sub-teams, but also the number of sub-teams that interact with other sub-teams (the parameter Y in the NKC model), especially when the number of sub-teams is small.

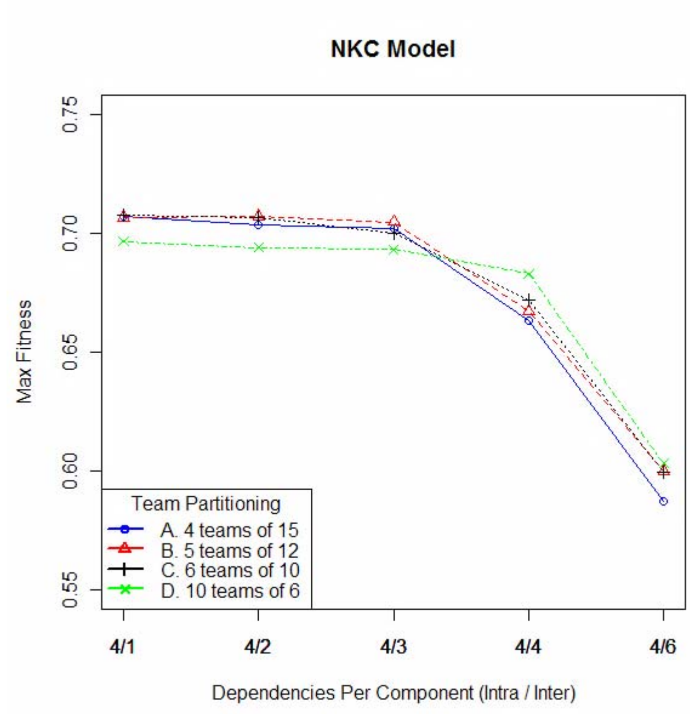

Figure 21. Comparison of results for overall project fitness achieved for identically sized projects that are partitioned differently.

Project Dependency Parameters	**Dependencies**			**Avg Max Fitness**				**Significance Test**
	Total	*Per Component*	*Intra/Inter*	*A. S=4, N=15*	*B. S=5, N=12*	*C. S=6, N=10*	*D. S=10, N=6*	*HSD Sig.*
K'=4, C=1, Y=1	300	5	4/1	0.7072	0.7066	0.7075	0.6966	A-D, B-D, C-D
K'=4, C=2, Y=1	360	6	4/2	0.7036	0.7072	0.7064	0.694	A-D, B-D, C-D
K'=4,C=3, Y=1	420	7	4/3	0.7022	0.7047	0.7002	0.6932	A-D, B-D
K'=4, C=2, Y=2	480	8	4/4	0.663	0.6671	0.672	0.683	A-D, B-D
K'=4, C=2, Y=3	600	10	4/6	0.587	0.6001	0.5994	0.6032	A-B,A-C, A-D

Table 11. Numerical results for average maximum fitness in identically sized projects with different partitioning of teams. Each line in the table reports a comparison of results in which the parameters N and S were varied, giving some projects fewer larger teams and other projects a greater number of smaller teams.

From Figure 21, we find that while there are some differences in fitness due to project partitioning, a more notable difference in fitness results when we modify the number of inter-team dependencies while holding the intra-team dependencies constant (K'=4). This result is the same as we observed in Experiment 1, but we note that the result remains the same regardless of the project partitioning scheme. That is, projects with fewer large teams and many small teams experience the same phenomenon: better performance when we increase the internal coupling of interdependencies within sub-teams, relative to coupling between sub-teams. This suggests that overall system fitness in co-evolving groups is more sensitive to changes in the structure of inter-team and intra-team coupling of interdependencies than changes in how the system is partitioned into groups.

Nash Equilibrium (Development Time)

Figure 22 and

Table 12 reveal that our results for the time to reach Nash equilibrium in a system are not unlike those obtained for overall system fitness in this experiment. That is, when we partition projects with an equal number of interdependencies using different schemes, the pattern of results remains the same with few significant differences between partitioning schemes. As with overall system fitness, what does seem to matter is the "density" of interdependencies within each individual sub-team relative to the number of inter-team dependencies. Specifically, when we create sub-teams that have a higher degree of internal dependency relative to external dependencies, we find that the time to reach Nash equilibrium is reduced. This finding suggests that "modularizing" the project, by creating subsystems with a higher density of interdependencies, yields a significant advantage in terms of development time for NPD projects. Conversely, we note in Figure 22 that there is a convergence to 2,500 steps which is the value at which we set the *max steps* variable in this experiment. In this experiment, the additional dependencies per component are all external to the subsystem of interest. Thus, in reality, this means that as we increase the ratio of external to internal dependencies, it is likely that more projects may not converge to a design solution.

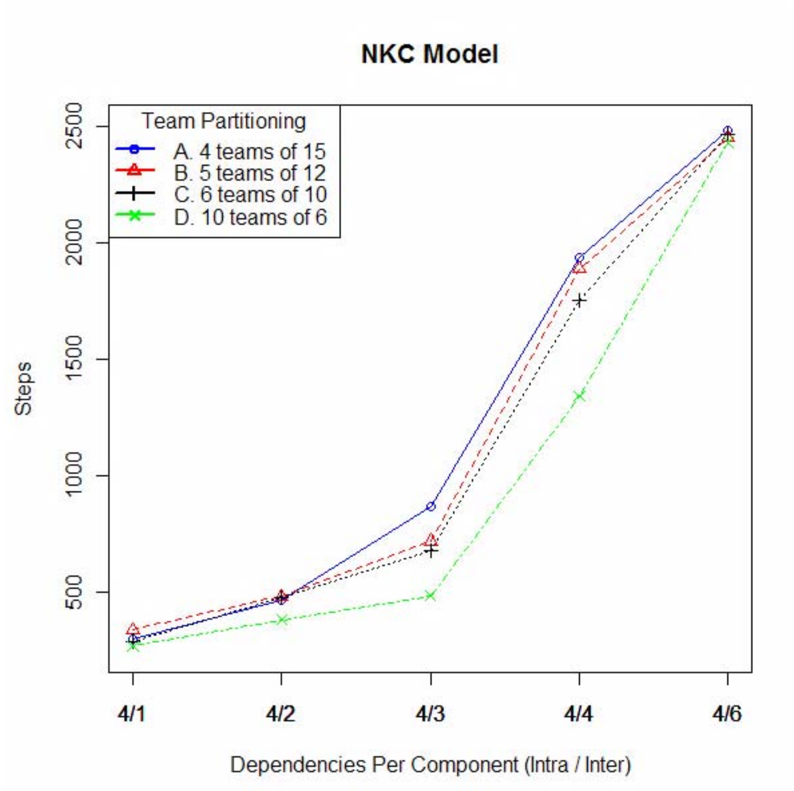

Figure 22. Comparison of results for time to reach Nash equilibrium for identically sized projects that are partitioned differently.

Project Dependency Parameters	**Dependencies**			**Time to Reach Nash Equilibrium**				**Significance Test**
	Total	*Per Component*	*Intra/Inter*	*A. S=4, N=15*	*B. S=5, N=12*	*C. S=6, N=10*	*D. S=10, N=6*	*HSD Sig.*
K=4, C=1, Y=1	300	5	4/1	299.4	340.1	292.1	271.4	B-D
K=4, C=2, Y=1	360	6	4/2	467.1	483.5	478.4	383.4	None
K=4,C=3, Y=1	420	7	4/3	869.2	720.7	679	485.4	A-D, B-D, C-D, A-B, A-C
K=4, C=2, Y=2	480	8	4/4	1937	1888	1755	1341	A-D, B-D, C-D
K=4, C=2, Y=3	600	10	4/6	2483	2450	2466	2430	None

Table 12. Numerical results for average time to reach Nash equilibrium in identically sized projects with different partitioning of teams. Each line in the table reports a comparison of results in which the parameters N and S were varied, giving some projects fewer larger teams and other projects a greater number of smaller teams.

The benefit of shorter development times for projects with a higher degree of modularity (high internal coupling and low external coupling of dependencies) is explained as follows. When the design decisions *within* a sub-team are highly interdependent, the fitness landscape of each sub-team becomes more rugged which leads to shorter adaptive walks for each team because the probability of any given team finding a local optimum at a given step increases. Additionally, due to the higher peaks in the rugged coupled landscapes, each team is also more likely to find a higher local optimum which explains why the overall system fitness is improved, on average, when internal dependency (K') is increased.

One notable difference between project partitioning schemes does exist, as shown in Figure 22: for constant parameters K', C, and Y, projects with a greater number of teams consistently take less time to reach Nash equilibrium than fewer teams. This result is somewhat surprising, as we had initially thought that a greater number of teams would be associated with greater coordination and, thus, longer time to reach Nash equilibrium. However, we found that it is not solely the number of teams that contribute to the development time of a project; instead, a competing consideration—team size—also plays a critical role in the time to reach Nash equilibrium. Specifically, the smaller teams have an advantage because the size of their search space is orders of magnitude smaller than that of larger teams. We recall that in the NK model the total number of possible configurations equals 2^N. So, in an NPD context of the NKC model, smaller sub-teams can search their individual fitness landscapes and reach a local maximum faster, but there are more co-evolving teams that threaten to alter the search of each sub-team which prolongs the search. Yet, in the end, given a constant project size, our results suggest that smaller sub-team size can yield an advantage in terms of development time, despite the necessity of having to coordinate a greater number of sub-teams. To our knowledge, this result, nor its quantification via simulation, has been reported in the extant literature.

5.4.3. Experiment 3: Inter-team vs. Intra-team design interdependencies

In this experiment, we more fully explore how NPD project architectures impact project performance measures. Our results, to this point, suggest that benefits accrue to NPD projects when design decision interdependencies are contained within each team, as opposed to between teams. NPD projects with this characteristic of high internal coupling of interdependencies are said to have a modular architecture. We simulate a relatively large number (113) of stylized NPD project interaction structures by varying the parameters S, N, K, C, and Y in the NKC model. For each of the stylized NPD project architectures, we conducted 100 independent runs of the NKC model simulation.

Figure 23 plots the average maximum fitness and the time to reach Nash equilibrium against a measure of project modularity which we define as the ratio of intra-team interdependencies to inter-team interdependencies[13]. More formally, in terms of the parameters in the NKC model we define this measure as the interdependency ratio R:

$$R = \frac{K'}{C \times Y}$$

which gives the number of components *within* each team that influence a given component in comparison to the number of components from *other* teams that influence that same component. Thus, a higher ratio is an indicator of higher design dependency coupling internal to each team which is associated with a higher degree of modularity[14].

[13] The table of numerical results is not printed in the main text of this article to conserve space, but is available in Appendix A.

[14] Several measures of product modularity have been proposed in the literature (Sosa et al., 2003; Hölttä-Otto and de Weck, 2007), and in this study our measure of modularity is consistent with the notion that modular products have high internal coupling within each module.

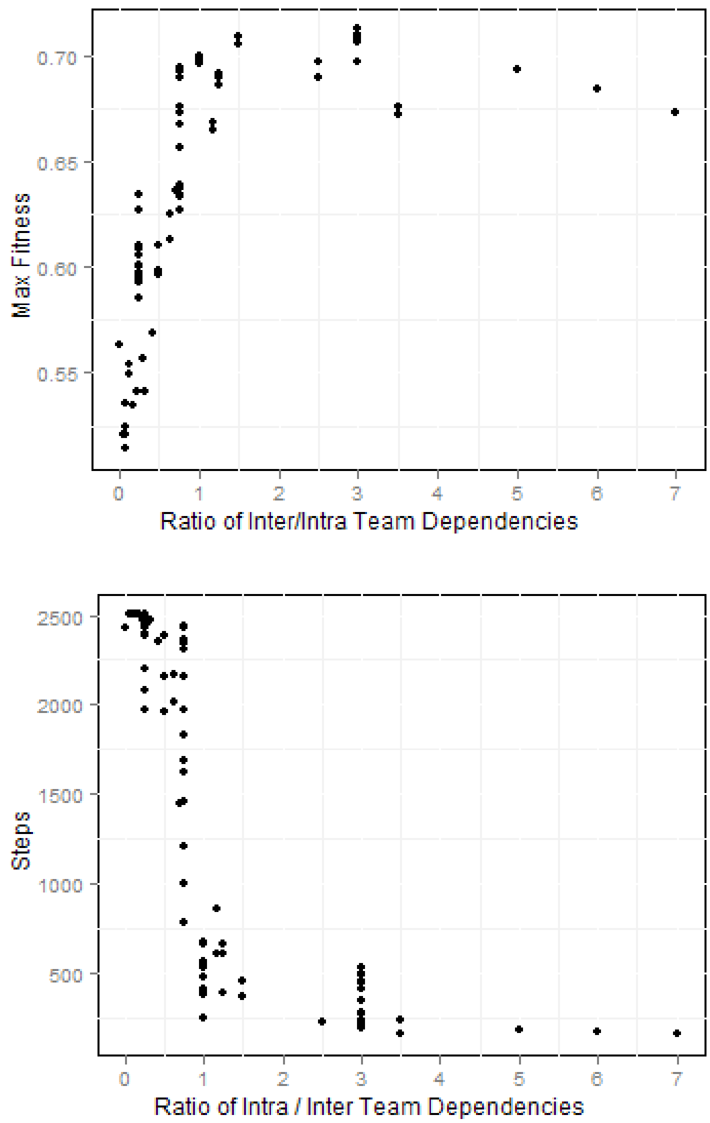

Figure 23. Average maximum fitness and time to reach Nash equilibrium, plotted as a function of project modularity.

In Figure 23 it can clearly be seen that, as modularity increases, there is a rapid rise in the average maximum fitness achieved. However, increased modularity appears to have not only an upper limit to its benefit, but also a negative effect on fitness achieved when projects become extremely modular. As projects become very modular, there are few (if any) interdependencies among the sub-teams and the subsystems for which they have responsibility. Under most conditions, interdependencies among sub-teams and modules are not desired because they reduce overall system fitness by constantly frustrating each sub-team's adaptive search. Yet, interdependencies among sub-teams also create opportunities for synergistic, or complementary, design decisions. So, when these opportunities become overly sparse or non-existent, overall system fitness is lower than at levels of more moderate modularity. Other researchers have warned that excessive modularity can lead to ignorance of important cross-module interactions (Ethiraj & Levinthal, 2004). Our study, however, quantifies this caution through empirical simulation results using the NKC model which has previously seen limited application.

Figure 23 also shows that projects with a very low degree of modularity ($R < 1$) take far greater time to reach Nash equilibrium, and many projects do not converge to a solution within the time allotted (2,500 simulation steps) in this experiment. In Figure 23, there does, however, appear to be a critical value of modularity ($R \sim 0.75$), above which projects experience a profound improvement in the time required to reach Nash equilibrium. It should be noted that when R=0.75, there is high variability in the time to reach Nash equilibrium. This high variability is attributed to the size of each sub-team or the number of subteams: projects with smaller teams or fewer teams perform better than projects with larger sub-teams or more sub-teams, for a constant value of R. For example, in this study an R value of 0.75 results when K=3 while C=2 and Y=2. When sub-teams are smaller (e.g. N=5), a constant value of K=3 yields dense set of internal interdependencies for each sub-team which, as previously noted, provides an advantage in terms of development time (and overall system fitness). One implication is that if the project will be partitioned such that each team will have a relatively fixed number of interdependencies for each component in each sub-team, then smaller teams are preferred to large teams and fewer teams are preferred to more teams. This effect can be seen graphically in Figure 21 and Figure 22 from Experiment 2.

In sum, it remains that the central result of this experiment is the suggestion that increased modularity in NPD projects has significant benefits in terms of both overall system fitness and development time—up to a point. The benefit to overall system fitness levels off and then actually regresses as modularity becomes extreme, due to the sparseness of potentially complementary interdependencies. In terms of development time, the benefit of increased modularity levels off but does not regress. These results, taken together, imply that moderate modularity may, in fact, be preferred to completely modular designs.

5.4.4. Experiment 4: A Better Search Strategy?

Having modeled NPD as a co-evolutionary process in which the decisions and actions of sub-teams influence the performance of one another, often creating an unstable process that does not converge well, our final experiment briefly explores how modifying the behaviors of sub-teams might improve this situation. Specifically, we impose a rule regarding whether to adopt the result of a design change which states: changes will only be accepted if they yield greater than 5 percent improvement for the subsystem. This assumption is not unrealistic in the domain of NPD in which small marginal improvements are likely not worth the associated time and cost to implement.

Figure 24 plots a comparison of overall system fitness and time to reach Nash equilibrium for highly modular systems with the following NKC model parameters: S=5, N=8, K=5, C=1, Y=1 (R=5). In this comparison, it can be observed that when we do not accept small marginal improvements, the average maximum fitness is slightly reduced because some paths to higher overall system fitness are not explored. However, we also find that there is a reduction in the time to reach Nash equilibrium because very small improvements to subsystems, which often negatively affect other subsystems in a disproportionate fashion, are not adopted.

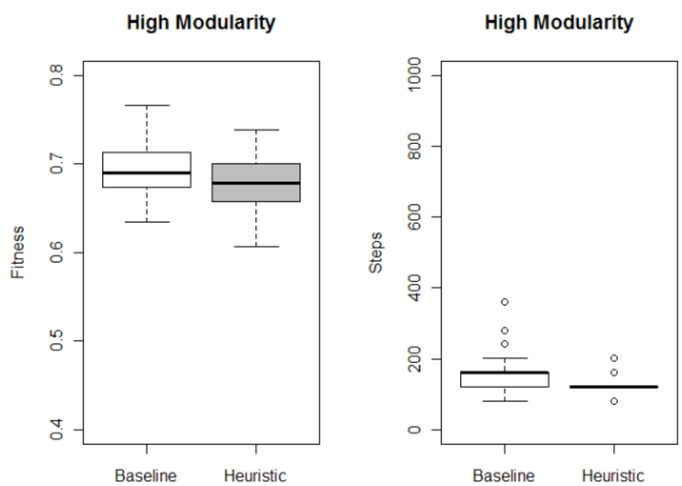

Figure 24. Results for overall project fitness and time to reach Nash equilibrium when very small subsystem fitness gains are not adopted in projects with high modularity. These plots show a comparison of this decision heuristic with the results obtained using the standard rule where *any* fitness improvement at the subsystem level is adopted.

Likewise, we modeled identical low modularity systems, using both the standard NKC rules and our modified heuristic, with the following model parameters: S=5, N=8, K=3, C=3, Y=2 (R=0.5). For NPD projects with low modularity,Figure 25 shows profoundly different results. First, average maximum fitness of projects is markedly improved by not adopting small marginal improvements to subsystems. This result is explained by the fact that, in low modularity systems (in comparison to high modularity systems), there are many interdependencies among subsystems and, therefore, the adoption of a change in one subsystem typically leads to changes of greater magnitude in a higher number of other subsystems. By not adopting small marginal improvements, we effectively limit the number of times that interdependent subsystems are perturbed or "knocked off" their own incremental improvement paths. Sub-teams are, therefore, able to locate better overall design configurations because they are less "interrupted" by the decisions of other sub-teams. The time to reach Nash equilibrium in low modularity systems is also benefitted by not adopting small marginal subsystem improvements because sub-teams are only affecting the fitness of other sub-teams when a marginal improvement is more beneficial, as opposed to generating rework across the many parts of the project in the pursuit of trivial subsystem improvements.

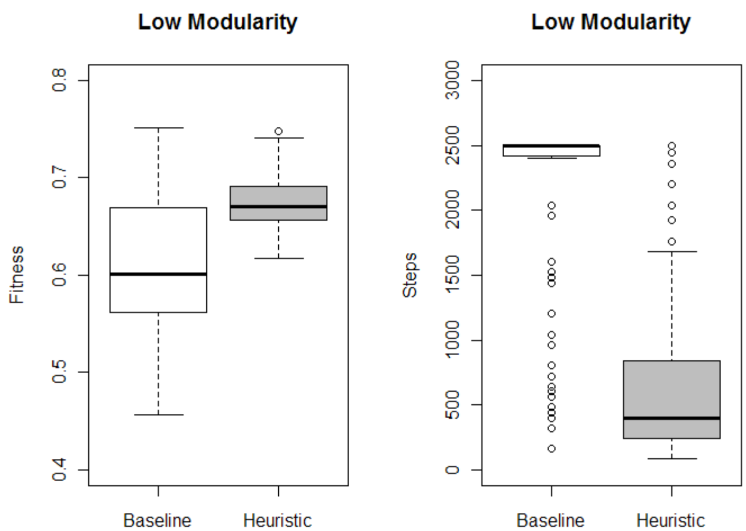

Figure 25. Results for overall project fitness and time to reach Nash equilibrium when very small subsystem fitness gains are not adopted in projects with low modularity. These plots show a comparison of this decision heuristic with the results obtained using the standard rule where *any* fitness improvement at the subsystem level is adopted.

In sum, the results of this experiment suggest that, in general, rules regarding individual sub-team decisions can have a significant impact on the outcomes of an NPD project. Specifically, by not adopting trivial subsystem improvements, highly modular systems experience a small fitness decrease, but a corresponding improvement in development time. On the other hand, in low modularity systems, the elimination of trivial improvement adoption significantly benefits both the fitness of the system *and* the time to develop the system.

5.5. Discussion

The application of complexity models to organizational decision making and search problems has a well-defined literature. The NK model has often been the focus of such research, while the NKC model has received far less attention and exploration, partly due to its computational complexity, and partly due to its lack of use within the research community. Using both the NK and NKC models, we show that the NKC model yields significantly different results for systems with identical interdependency structures. From an NPD modeling perspective, this has an important implication: if an organization being modeled is characterized by sub-teams that make local decisions on their own fitness landscape, but whose decisions also influence other sub-teams, then the NKC model provides a more realistic representation of such organizations. In the domain of NPD, we believe this is the case and that the NKC model should be given serious consideration over the more popular NK model.

How to divide or partition the total design effort in large NPD projects is an important strategic consideration. For example, should there be smaller number of large design teams, or a higher number of small design teams? We found that this tradeoff between team size and number of teams is largely offset as long as the numbers of internal interdependencies and external interdependencies are held constant, though our results do suggest that smaller sub-teams result in faster development times, but that smaller sub-teams also yield inferior overall project fitness under conditions of few external sub-team dependencies. One major implication for managers of large design projects, based on this research, concerns the ratio of internal interdependencies to external interdependencies for each sub-team or subsystem. Using the NKC model, we showed that projects with highly interdependent sub-teams tend to perform poorly, both with respect to overall fitness of the project and the time to reach a solution. We showed that the performance of such projects can be significantly improved by adopting more modular interdependency structures, in which each team has a higher ratio of internal to external dependencies. Extremely modular projects, however, run the risk of degraded fitness due to elimination of potentially beneficial complementary interdependencies. It has also been shown, previously, that very modular systems are subject to imitation by competitors (Rivkin, 2000).

In some design projects, however, managers may not have total control over how teams and design interdependencies are structured. If this is the case, we showed that outcomes can be influenced by altering the decision criteria for adopting design changes. For highly modular systems, eliminating trivial marginal improvements comes with a tradeoff, in general, between lower fitness and improved time to reach a solution. Perhaps more importantly, if a design effort is characterized by many interdependent sub-teams, significant improvements can be made by not allowing individual design teams to adopt design changes that only yield small marginal improvements to their own subsystem.

5.6. Conclusion

Ultimately, the study and modeling of co-evolutionary design teams is important, especially with the advent of virtual organizations that are geographically dispersed and more autonomous. In this paper, we have begun to examine how co-evolutionary team dynamics, governed by a relatively small number of parameters, can influence the performance of NPD projects. We revealed that the nature of many NPD projects—specifically, those in which design efforts are divided among several teams—demands a different modeling framework than one that assumes centralized management and decision making. The results of our simulations also help make an important distinction regarding complexity in NPD: it is not so much the size of the NPD project, or the density of dependencies within a project, that drives complexity. In fact, we found that, in projects of identical size, some projects with more dependencies outperformed those with fewer dependencies. This is because complexity in projects characterized by subdivision of design tasks is driven by how the dependencies are distributed within and among the work teams.

Our research in this study opens several new and important research directions. First, we only explored one possible decision rule modification, but important insights could be gained by exploring how alternate search behaviors among co-evolutionary teams impact project-level outcomes. For instance, what if some design activities are "frozen" after they reach a threshold level of performance? Or, what if teams and agents engage in satisficing behavior, rather than constantly seeking higher fitness? Second, our modeling used stylized project and team interdependency structures, in which we assumed homogeneous values for team size (N), internal complexity (K), and external influences (C and Y) within each simulated project. However, many more patterns of interaction exist for real-world projects; namely, in many projects some components or subsystems have many more interdependencies than others and, thus, can either be more sensitive to changes or cause more perturbations within the system. Rivkin and Siggelkow (2007) explore several other stylized patterns of interdependencies, but only for one team at a time. Investigation into how co-evolutionary teams with different interdependency structures would add to the research on NPD modeling. Additionally, in NPD projects, managers and engineers typically have some knowledge regarding how a change to one component will affect other components. Therefore, structured (Solow et al., 1999) and weighted interdependencies (e.g. Pimmler and Eppinger 1994) could be modeled and explored in the context of co-evolutionary design projects.

Understanding complex NPD projects from the perspective of interdependent design teams requires a modeling framework that accounts for co-evolutionary design decisions. It is our hope that we have begun to pave a path of new research that uses the NKC model in the NPD domain by providing insight into 1) how NPD projects can be modeled, 2) the trade-offs among different approaches to organizational and product architectures, and 3) future research paths stemming from this current study.

Chapter 6: Conclusions and Future Research Directions

The NK model has provided numerous researchers in diverse disciplines with a tool for modeling the interaction of entities at multiple scales, from genes to people to firms to industries. A review of the literature on new product development (NPD) suggests that the NK model may be well-suited for modeling the interacting decisions among designers and the dependencies between components that give rise to complexity in the NPD process. However, a critical review of the NK model finds that the NK model can, and should, be modified if it is be used in NPD research and practice.

The primary contribution of this dissertation is the detailed investigation and treatment of 1) the assumptions of the NK model and 2) the contextual realities of NPD. We integrate these two relevant areas of investigation and argue that the NK model can be extended in a number of different ways to reflect real-world NPD dynamics. Our research has led us to develop new modeling constructs for applying the NK model to the NPD process, as well as overarching insights for NPD practice. In this summary chapter, we synthesize and present the salient the findings from the three related research efforts that comprise this dissertation, and identify promising future research directions for NPD stemming from this research.

6.1 Overarching Findings and Insights

Our first study in this dissertation examined search behavior in NPD, as well as costs associated with search. Next, we explicitly explored the modeling of interactions between decisions and components within an NPD project. Finally, we investigated the nature of how NPD projects are generally organized and managed, and concerned ourselves with questions related to span of control for teams and how teams interact internally as well as with other teams on the same NPD project. We now highlight the primary conclusions and contributions from our research:

- Viewing NPD as a search problem, we discuss that the NK model does not attribute any cost to each move on the fitness landscape. We also discuss that search agents in NK model do not pursue moves fail to yield an overall system improvement. We argue that these two notions are not congruent with how NPD firms and teams actually operate. This is an important theoretical insight if, in fact, NPD researchers are to use NK model. From a practical perspective our extended NK model that relaxes these two assumptions then finds that firms faced with increased complexity stand to benefit most from increased exploration of paths that, at first, do not appear promising. However, we also find that the benefit of increased exploration experiences diminishing returns.
- After examining the search behavior of agents and firms, we analyze how dependencies and component-level changes in the NK model align with the realities of NPD. We find that there are additional assumptions of the NK model that are not congruent with NPD practice. Specifically, we note that NPD managers and engineers typically have some knowledge regarding the dependency between system elements. We also note that NPD managers and engineers make conscious (not random) decisions regarding component-level changes which necessitate a different probability distribution to be applied in the modeling of changes. Along these lines, we develop two new modeling constructs that extend the NK model for NPD and produce managerial insights. Namely, whereas the NK model calls attention to the number dependencies within a system, we find that the nature of dependencies between components is a more important determinant of project outcomes.
- In our final study, we explicitly model structures typical of how NPD firms organize and manage projects. For NPD modeling and research, we find that the NK model alone does not adequately represent how decisions and teams interact on projects which have multiple teams that have some level of interdependency between them. Instead, a variant of the NK model, known as the NKC model, generates a federation of linked NK models which more realistically captures the dynamic of co-evolving teams on the same project. This is an important theoretical consideration because identical dependency structures can yield profoundly different results under the NK model versus the NKC model. Additionally, from an applied perspective, we explored the trade-off between partitioning a project into

fewer large teams versus more small teams. We found that for a constant project size and constant number of dependencies per component, a larger number of smaller teams is able to converge to a solution faster, on average, than fewer large teams but at some expense in terms of overall project performance when the project is characterized by modularity. We also found that long search processes in projects with co-evolving teams can be attenuated by modifying the team level behavior; specifically, if teams forego trivial subsystem-level improvements, the overall search process is greatly reduced, especially for projects with highly interconnected teams.

6.2 Future Research Directions

The investigations and explorations into modeling NPD as a complex system using the NK modeling framework lead us to a series of promising future research directions. Generally speaking, because the NK and NKC models are agent-based models, we believe that some of the most promising future research lies in agent-level modeling to produce system level insights. Specifically:

1. In this research we have modeled agents using simple behavior rules, but decision makers often behave in unforeseen ways as they are subject to many different motives and incentives. Therefore, we feel that building a participatory simulation (e.g. Colella, 2000) can help reveal how decision-makers behave under different conditions and situations which, in turn, will allow for building more realistic models of NPD.
2. Related to the notion of decision maker behaviors, the research in this dissertation reveals that teams acting locally can adversely affect the performance of the overall project, especially when teams have a high degree of interdependence. Therefore, research into structures and strategies that can incentivize managers and engineers to act with an enhanced project-level perspective has significant value.
3. Finally, an important research direction relates to empirically validating the impact of complexity in NPD projects. Finding primary sources of data on NPD project decisions and performance is admittedly difficult due to the confidential nature of many such data. However, some firms and organizations may be willing to provide such data if the data is masked such that it is not attributable to a given firm or project, if they can be convinced of the significant potential benefits from modeling and simulation of complex NPD projects.

Additionally, it is recognized that the modeling and simulation approach used to explore NPD dynamics in this research is but one possible approach. It was shown that the dependencies and interactions within NPD projects can be represented using a matrix form. Therefore, analytical modeling approaches, using concepts such as principal component analysis, could potentially be used to gain additional insights regarding the structure of large NPD projects

Complexity science is a nascent science, not yet fully developed or appreciated. New product development is also a relatively new endeavor in the course of human history. Therefore, it is critical that we better understand how networks of human activity and decision making impact the outcome of projects that directly impact firm competitiveness and, ultimately, our competitiveness as a nation in the global economy.

Appendix A

In this appendix we present the full table of results for our experiment from Section 5.4.3, in which we explored how project product architectures impact project performance measures. We recall that R is the ratio of intra-team dependencies to inter-team dependencies, defined formally as:

$$R = \frac{K'}{C \times Y}$$

where K' is the number of dependencies, per component, within each team; C is the number of other sub-teams that create at least one dependency for each component; Y is the number of dependencies that each of the C teams create for each component. Therefore, the product of C and Y gives the total number of external dependencies for each component.

S	N	K	C	Y	Total Dependencies	Modularity, R	Avg. Max Fitness	Avg. Time to Nash equlilibrium
5	8	0	5	1	200	0.00	0.5631	2433
5	8	1	5	4	840	0.05	0.5212	2501
5	8	1	4	4	680	0.06	0.5211	2501
5	8	1	8	2	680	0.06	0.514	2501
5	8	1	3	4	520	0.08	0.5247	2501
5	8	1	4	3	520	0.08	0.5356	2501
5	8	1	2	4	360	0.13	0.5538	2501
5	8	1	4	2	360	0.13	0.5499	2501
5	8	3	6	3	840	0.17	0.5351	2501
5	8	3	7	2	680	0.21	0.5416	2471
5	8	1	4	1	200	0.25	0.634	1966
5	8	1	1	4	200	0.25	0.6055	2398
5	8	1	2	2	200	0.25	0.6269	2199
5	8	1	2	2	200	0.25	0.6269	2199
8	5	1	2	2	200	0.25	0.6342	2077
6	8	1	2	2	240	0.25	0.6083	2396
8	6	1	2	2	240	0.25	0.6102	2384
7	8	1	2	2	280	0.25	0.6	2430
8	7	1	2	2	280	0.25	0.5979	2453
8	8	1	2	2	320	0.25	0.596	2468
8	8	1	2	2	320	0.25	0.596	2468
8	9	1	2	2	360	0.25	0.5973	2488
9	8	1	2	2	360	0.25	0.5948	2494
8	10	1	2	2	400	0.25	0.5928	2487
10	8	1	2	2	400	0.25	0.5978	2501
8	11	1	2	2	440	0.25	0.5854	2501
11	8	1	2	2	440	0.25	0.6012	2487
5	8	3	5	2	520	0.30	0.5568	2457
5	8	5	8	2	840	0.31	0.5415	2471
5	8	5	6	2	680	0.42	0.5691	2351
5	8	3	3	2	360	0.50	0.6104	2159
5	8	3	2	3	360	0.50	0.5964	2389
5	8	7	7	2	840	0.50	0.5981	1957
5	8	5	4	2	520	0.63	0.6247	2019
5	8	5	2	4	520	0.63	0.6131	2169
4	15	4	2	3	600	0.67	0.587	2483
5	12	4	2	3	600	0.67	0.6001	2450
6	10	4	2	3	600	0.67	0.5994	2466
10	6	4	2	3	600	0.67	0.6032	2430

S	N	K	C	Y	Total Dependencies	Modularity, R	Avg. Max Fitness	Avg. Time to Nash equlilibrium
5	8	7	5	2	680	0.70	0.6366	1446
5	8	3	4	1	280	0.75	0.6923	784.6
5	8	3	1	4	280	0.75	0.6675	1621
5	8	3	2	2	280	0.75	0.6899	1209
8	5	3	2	2	280	0.75	0.6944	995.4
6	8	3	2	2	336	0.75	0.6735	1691
8	6	3	2	2	336	0.75	0.6756	1456
7	8	3	2	2	392	0.75	0.6675	1824
8	7	3	2	2	392	0.75	0.6562	1966
8	8	3	2	2	448	0.75	0.6386	2160
8	8	3	2	2	448	0.75	0.6386	2160
8	9	3	2	2	504	0.75	0.6272	2366
9	8	3	2	2	504	0.75	0.6382	2312
8	10	3	2	2	560	0.75	0.6335	2433
10	8	3	2	2	560	0.75	0.6367	2337
8	11	3	2	2	616	0.75	0.6273	2435
11	8	3	2	2	616	0.75	0.6345	2366
5	8	1	1	1	80	1.00	0.6994	242
8	5	1	1	1	80	1.00	0.696	479.6
6	8	1	1	1	96	1.00	0.6968	413.2
8	6	1	1	1	96	1.00	0.7001	399.2
7	8	1	1	1	112	1.00	0.6961	381.2
8	7	1	1	1	112	1.00	0.7002	395.7
8	8	1	1	1	128	1.00	0.6989	529.4
8	9	1	1	1	144	1.00	0.6958	561.3
9	8	1	1	1	144	1.00	0.6999	481.5
8	10	1	1	1	160	1.00	0.6985	666
10	8	1	1	1	160	1.00	0.6996	544.4
8	11	1	1	1	176	1.00	0.6978	534.8
11	8	1	1	1	176	1.00	0.6972	670.8
4	15	4	2	2	480	1.00	0.663	1937
5	12	4	2	2	480	1.00	0.6671	1888
6	10	4	2	2	480	1.00	0.672	1755
10	6	4	2	2	480	1.00	0.683	1341
5	8	7	3	2	520	1.17	0.6651	610
5	8	7	2	3	520	1.17	0.6687	858.8
5	8	5	1	4	360	1.25	0.6913	664.6
5	8	5	4	1	360	1.25	0.6861	394.4
5	8	5	2	2	360	1.25	0.6896	607.8
4	15	4	3	1	420	1.33	0.7022	869.2

S	N	K	C	Y	Total Dependencies	Modularity, R	Avg. Max Fitness	Avg. Time to Nash equlilibrium
5	12	4	3	1	420	1.33	0.7047	720.7
6	10	4	3	1	420	1.33	0.7002	679
10	6	4	3	1	420	1.33	0.6932	485.4
5	8	3	2	1	200	1.50	0.7053	362.4
5	8	3	1	2	200	1.50	0.7094	453
4	15	4	2	1	360	2.00	0.7036	467.1
5	12	4	2	1	360	2.00	0.7072	483.5
6	10	4	2	1	360	2.00	0.7064	478.4
10	6	4	2	1	360	2.00	0.694	383.4
5	8	5	2	1	280	2.50	0.6893	222
5	8	5	1	2	280	2.50	0.6973	224.2
5	8	3	1	1	160	3.00	0.7101	214.2
8	5	3	1	1	160	3.00	0.697	191.2
6	8	3	1	1	192	3.00	0.7083	237.7
8	6	3	1	1	192	3.00	0.706	274.2
7	8	3	1	1	224	3.00	0.707	278.6
8	7	3	1	1	224	3.00	0.7078	283
8	8	3	1	1	256	3.00	0.7124	341.5
8	8	3	1	1	256	3.00	0.7124	341.5
8	9	3	1	1	288	3.00	0.7093	410.5
9	8	3	1	1	288	3.00	0.7075	450.2
8	10	3	1	1	320	3.00	0.7074	492
10	8	3	1	1	320	3.00	0.709	443
8	11	3	1	1	352	3.00	0.7093	496.3
11	8	3	1	1	352	3.00	0.707	526.6
5	8	7	1	2	360	3.50	0.6762	240.5
5	8	7	2	1	360	3.50	0.6722	160.6
4	15	4	1	1	300	4.00	0.7072	299.4
5	12	4	1	1	300	4.00	0.7066	340.1
6	10	4	1	1	300	4.00	0.7075	292.1
10	6	4	1	1	300	4.00	0.6966	271.4
5	8	5	1	1	240	5.00	0.6936	181.6
5	8	6	1	1	280	6.00	0.6838	167.2
5	8	7	1	1	320	7.00	0.6733	154.8

References

Altenberg, L. (1997). NK fitness landscapes. In *Handbook of Evolutionary Computation*. New York: Oxford University Press.

Anderson, P. (1999). Perspective: Complexity theory and organization science. O*rganization Science*, *10*(3), 216–232.

Baldwin, C. Y., & Clark, K. B. (2000). *Design Rules, Volume 1: The power of modularity* (Vol. 1). MIT Press.

Bar-Yam, Y. (2003). When systems engineering fails-toward complex systems engineering. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on* (Vol. 2, pp. 2021–2028).

Battiti, R., Brunato, M., & Mascia, F. (2008). *Reactive search and intelligent optimization* (Vol. 45). Springer.

Baumann, O., & Siggelkow, N. (2012). Dealing with Complexity: Integrated vs. Chunky Search Processes. *Organization Science*, *Articles in Advance*. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1104617

Billinger, S., Stieglitz, N., & Schumacher, T. (2010). Search on rugged landscapes: An experimental study. *Available at SSRN 1711743*.

Black, T. A., Fine, C. H., & Sachs, E. M. (1990). A method for systems design using precedence relationships: An Application to automotive brake systems. *MIT Sloan School of Management, Working Paper no. 3208*.

Brown, S. L., & Eisenhardt, K. M. (1998). *Competing on the Edge: Strategy as Structured Chaos*. Harvard Business Press.

Browning, T. R., Fricke, E., & Negele, H. (2006). Key concepts in modeling product development processes. *Systems Engineering*, *9*(2), 104–128.

Chao, R. O., & Kavadias, S. (2008). A theoretical framework for managing the new product development portfolio: when and how to use strategic buckets. *Management Science*, *54*(5), 907–921.

Chen, W.-R., & Miller, K. D. (2007). Situational and institutional determinants of firms' R&D search intensity. *Strategic Management Journal*, *28*(4), 369–381.

Clark, K. B., & Fujimoto, T. (1991). *Product development performance: Strategy, organization, and management in the world auto industry*. Harvard Business Press.

Colella, V. (2000). Participatory simulations: Building collaborative understanding through immersive dynamic modeling. *The journal of the Learning Sciences*, *9*(4), 471-500.

Cooper, R. G. (1990). Stage-gate systems: a new tool for managing new products. *Business Horizons*, *33*(3), 44–54.

Cooper, R. G. (1994). Perspective: Third-generation new product processes. *Journal of New Product Innovation Management*, *11*(1), 3–14.

Cooper, R. G., Edgett, S. J., & Kleinschmidt, E. J. (1999). New product portfolio management: practices and performance. *Journal of Product Innovation Management*, *16*(4), 333–351.

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

Daft, R. L. (2009). *Organization Theory and Design*. South-Western Pub.

Dixit, A. K., & Pindyck, R. S. (1994). *Investment Under Uncertainty*. Princeton, NJ: Princeton Univ Press.

Dooley, K. J. (1997). A complex adaptive systems model of organization change. *Nonlinear Dynamics, Psychology, and Life Sciences*, *1*(1), 69–97.

Eisenhardt, K. M. (1988). Agency-and institutional-theory explanations: the case of retail sales compensation. *Academy of Management Journal*, *31*(3), 488–511.

Elmaghraby, S. E. (1995). Activity Nets: A guided tour through some recent developments. *European Journal of Operational Research*, *82*(3), 383–408.

Ethiraj, S. K., & Levinthal, D. (2004). Modularity and innovation in complex systems. *Management Science*, *50*(2), 159–173.

Fontana, W. (1990). Functional self-organization in complex systems. In*: 1990 Lectures in Complex Systems*, Sante Fe Institute Studies in the Sciences of Complexity, Lecture Notes Vol. III. Nadel, L. and D. Stein (eds.), 407–426.

Forrester, J. (1958). A Major Breakthrough for Decision Makers. *Harvard Business Review*, *36*(4), 37–66.

Frenken, K. (2001). Modelling the organisation of innovative activity using the NK-model. In *Nelson-and-Winter Conference* (pp. 12–16).

Frenken, K. (2006). Technological innovation and complexity theory. *Economics of Innovation and New Technology*, *15*(2), 137–155.

Galbraith, J. (1973). *Designing Complex Organizations*. Boston, MA: Addison-Wesley.

Ganco, M., & Hoetker, G. (2009). NK modeling methodology in the strategy literature: Bounded search on a rugged landscape. *Research methodology in strategy and management*, *5*, 237–268.

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

Gell-Mann, M. (1995). *The Quark and the Jaguar: Adventures in the Simple and the Complex*. St. Martin's Griffin.

Ghemawat, P., & Levinthal, D. (2008). Choice interactions and business strategy. *Management Science*, *54*(9), 1638–1651.

Green, P. E., & Srinivasan, V. (1978). Conjoint analysis in consumer research: issues and outlook. *Journal of consumer research*, 103–123.

Greve, H. R. (2003a). Investment and the behavioral theory of the firm: evidence from shipbuilding. *Industrial and Corporate Change*, *12*(5), 1051–1076.

Greve, H. R. (2003b). A behavioral theory of R&D expenditures and innovations: Evidence from shipbuilding. *Academy of Management Journal*, *46*(6), 685–702.

Greve, H. R. (1998). Performance, aspirations, and risky organizational change. *Administrative Science Quarterly*, 58–86.

Gupta, A., Pawar, K. S., & Smart, P. (2007). New product development in the pharmaceutical and telecommunication industries: A comparative study. *International Journal of Production Economics*, *106*(1), 41–60.

Holland, J. (1996). *Hidden order: How adaptation builds complexity*. Basic Books.

Holland, J. H. (2006). Studying complex adaptive systems. *Journal of Systems Science and Complexity*, *19*(1), 1–8.

Hölttä-Otto K, de Weck O (2007) Metrics for Assessing Coupling Density and Modularity in Complex Products and Systems. *ASME Conference on Design Theory and Methodology* , Las Vegas, NV.

Jablin, F. (1987). Formal organization structure. In *Handbook of Organizational Communication* (pp. 389–419). Newbury Park, CA: Sage.

Jarratt, T. A., Eckert, C. M., Caldwell, N. H., & Clarkson, P. J. (2011). Engineering change: an overview and perspective on the literature. *Research in Engineering Design*, *22*(2), 103–124.

Kahneman, D., & Tversky, A. (2000). *Choices, values, and frames*. Cambridge University Press.

Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA.

Kauffman, S. A., & Johnsen, S. (1991). Coevolution to the edge of chaos: Coupled fitness landscapes, poised states, and coevolutionary avalanches. *Journal of Theoretical Biology*, *149*(4), 467–505.

Kauffman, S., & Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical Biology*, *128*(1), 11–45.

Knudsen, T., & Levinthal, D. A. (2007). Two faces of search: Alternative generation and alternative evaluation. *Organization Science*, *18*(1), 39–54.

Krishnan, V., & Ulrich, K. T. (2001). Product development decisions: A review of the literature. *Management Science*, *47*(1), 1–21.

Langlois, R. N. (2002). Modularity in technology and organization. *Journal of Economic Behavior & Organization*, *49*(1), 19–37.

Lansing, J. S. (2003). Complex adaptive systems. *Annual Review of Anthropology*, 183–204.

Lenox, M. J., Rockart, S. F., & Lewin, A. Y. (2007). Interdependency, competition, and industry dynamics. *Management Science*, *53*(4), 599–615.

Levinthal, D. A. (1997). Adaptation on rugged landscapes. *Management science*, *43*(7), 934–950.

Levinthal, D. A., & Warglien, M. (1999). Landscape design: Designing for local action in complex worlds. *Organization Science*, *10*(3), 342–357.

Levitan, B., Lobo, J., Schuler, R., & Kauffman, S. (2002). Evolution of organizational performance and stability in a stochastic environment. *Computational & Mathematical Organization Theory*, *8*(4), 281–313.

Lieberman, M. B., & Montgomery, D. B. (1988). First-mover advantages. *Strategic management journal*, *9*(S1), 41–58.

Lorenz, E. N. (1995). *The Essence of Chaos*. Routledge.

Lovallo, D., & Kahneman, D. (2003). Delusions of success. *Harvard business review*, *81*(7), 56–63.

Lovallo, D. P., & Sibony, O. (2006). Distortions and deceptions in strategic decisions. *McKinsey Quarterly*, *1*, 18.

Maier, M. W., & Rechtin, E. (2000). *The Art of Systems Architecting*. Boca Raton, FL: CRC Press.

March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science*, *2*(1), 71–87.

Marion, R. (1999). The edge of organization: Chaos and complexity theories of formal social systems. Sage Publications, Inc.

McCarthy, I. P., Tsinopoulos, C., Allen, P., & Rose-Anderssen, C. (2006). New product development as a complex adaptive system of decisions. *Journal of Product Innovation Management*, *23*(5), 437–456.

McKelvey, B. (1999). Avoiding complexity catastrophe in coevolutionary pockets: Strategies for rugged landscapes. *Organization Science*, *10*(3), 294–321.

McMaster, M. (1996). *The Intelligence Advantage: Organizing for Complexity*. Boston, MA: Butterworth-Heinemann.

Menzel, C., & Mayer, R. J. (2006). The IDEF family of languages. *Handbook on architectures of information systems*, 215–249.

Mihm, J., Loch, C., & Huchzermeier, A. (2003). Problem–Solving Oscillations in Complex Engineering Projects. *Management Science*, *49*(6), 733–750.

Milgrom, P., & Roberts, J. (1990). Rationalizability, learning, and equilibrium in games with strategic complementarities. *Econometrica*, *58*(6), 1255–1277.

Mitchell, M. (2011). *Complexity: A Guided Tour*. Oxford University Press, USA.

Morelli, A., & Van den Biggelaar, K. (2009). *Managing New Product Development and Innovation in Challenging Times: Driving high performance through profitable innovation*. Accenture Outlook.

Nicolis, G., & Prigogine, I. (1977). *Self-Organization in Non-Equilibrium Systems: From Dissipative Structure Formation to Order through Fluctuations*. New York: Wiley & Sons.

Oakley, M. (1984). *Managing Product Design*. Weidenfeld and Nicolson.

Ostrom, E. (2009). A general framework for analyzing sustainability of social-ecological systems. *Science*, *325*(5939), 419–422.

Ottino, J. M. (2003). Complex systems. *AIChE Journal*, *49*(2), 292–299.

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, *15*(12), 1053–1058.

Rind, D. (1999). Complexity and Climate. *Science*, *284*, 105–107.

Rivkin, J. W. (2000). Imitation of complex strategies. *Management Science*, *46*(6), 824–844.

Rivkin, J. W., & Siggelkow, N. (2002). Organizational sticking points on NK landscapes. *Complexity*, *7*(5), 31–43.

Rivkin, J. W., & Siggelkow, N. (2007). Patterned interactions in complex systems: Implications for exploration. *Management Science*, *53*(7), 1068–1085.

Sanchez, R., & Mahoney, J. T. (1997). Modularity, flexibility, and knowledge management in product and organization design. *IEEE Eng Manage Rev*, *25*(4), 50–61.

Schilling, M. A., & Steensma, H. K. (2001). The use of modular organizational forms: an industry-level analysis. *Academy of Management Journal*, 1149–1168.

Siggelkow, N., & Levinthal, D. A. (2005). Escaping real (non-benign) competency traps: Linking the dynamics of organizational structure to the dynamics of search. *Strategic Organization*, *3*(1), 85–115.

Shalizi, C.R. (2006) Methods and techniques of complex systems science: An overview, in *Complex Systems Science in Biomedicine,* pp. 33-114, T.S Deisboeck and J.Y. Kresh, Eds. Springer.

Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society*, 467–482.

Simon, H. A. (1969). *The Sciences of the Artificial*. Boston, MA: MIT Press.

Simon, H. A. (1979). Rational decision making in business organizations. *The American Economic Review*, 493–513.

Solow, D., Burnetas, A., Tsai, M.-C., & Greenspan, N. S. (1999). Understanding and attenuating the complexity catastrophe in Kauffman's NK model of genome evolution. *Complexity*, *5*(1), 53–66.

Sommer, S. C., & Loch, C. H. (2004). Selectionism and learning in projects with complexity and unforeseeable uncertainty. *Management Science*, *50*(10), 1334–1347.

Sosa, M. E., Eppinger, S. D., & Rowles, C. M. (2003). Identifying modular and integrative systems and their impact on design team interactions. *Journal of Mechanical Design*, *125*(Part 2), 240–252.

Steward, D. V. (1981). Design structure system: A method for managing the design of complex systems. *IEEE TRANS. ENG. MGMT.*, *28*(3), 71–74.

Strogatz, S. H. (2001). Exploring complex networks. *Nature*, *410*(6825), 268–276.

Tushman, M., & O'Reilly, C. O. (1996). Evolution and revolution: Mastering the dynamics of innovation and change. *California Management Review*, *38*(4), 8–30.

Ulrich, K. T., & Eppinger, S. D. (1999). *Product Design and Development* (2nd ed.). McGraw-Hill New York.

United States Government Accountability Office (GAO). (2009). *Assessments of selected weapon programs* (No. GAO-09-326SP).

Von Bertalanffy, L. (1968). *General System Theory: Toundations, Developments and Applications*. New York: George Braziller, Inc.

Waterman, R., Peters, T., & Phillips, J. (1980). Structure is not organization. *Business Horizons*, *23*, 14–26.

Wiener, N. (1948). *Cybernetics*. New York: J. Wiley.

Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution. In *Proceedings of the sixth international congress on genetics* (Vol. 1, pp. 356–366).