Modeling Technology Evolution Using Generalized Genotype–Phenotype Maps

C. Jason Woodard
School of Information Systems
Singapore Management University
jwoodard@smu.edu.sg

Eric K. Clemons
The Wharton School
University of Pennsylvania
clemons@wharton.upenn.edu

ABSTRACT

Modeling open-ended technological evolution is notoriously challenging. The most successful models to date have been grounded in specific domains such as electronic circuit design. This paper presents an alternative approach based on a generalization of Kauffman's *NK* model. In this approach, boundedly rational agents combine components into products and systems whose value is determined by a random fitness landscape in which components may vary in their pleiotropy, or the number of genotypic functions they enable. The authors are developing a family of agent-based models using this framework, the first of which explores the evolution of platform architectures. Preliminary results from this model show that platforms emerge most strongly under conditions of frequent but moderate environmental change or a moderate number of correlated market niches.

Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: Model Development

General Terms

Design, Economics

Keywords

Technology evolution, generalized NK model, agent-based modeling, computational simulation

1. INTRODUCTION

The relentless proliferation of complex artifacts is a fact of everyday life in a modern economy, yet one that economics as a discipline remains hard-pressed to explain [9]. Among recent scholarship that has sought to account for both the existence of technological complexity and its explosive growth (e.g., [4]), a consensus is emerging that evolutionary theory offers a powerful way forward. It has been argued that "the economy is not like an evolutionary system, it is an evolutionary system" [10], which in turn suggests that technological evolution can be understood using the algorithmic tools of evolutionary computation [12].

This paper presents a framework for modeling the evolution of complex products and systems using computational agent-based research methods. Previous efforts have succeeded in specific domains like circuit design [5], but our goal is (loosely) to see how far we can get in a more general setting. Our main contribution is to extend Kauffman's *NK* model [23, 24, 26] to capture the key features of technology evolution in a market economy, building on prior work by Altenberg [2, 3] and Frenken [18].

We also provide a status report on an ongoing project to study the evolution of platform architectures using the proposed framework. Platform architectures are ubiquitous in complex engineered systems because they support both the reuse of highly interdependent "core" components and the combination of these components with a wide variety of loosely coupled "peripheral" components [8]. Preliminary results from this project indicate that these architectural properties can arise through simple evolutionary dynamics—without planning, learning, anticipation, or explicit design. The results also highlight the effects of two external conditions: the degree of environmental stability or change, and the degree of uniformity or heterogeneity among consumers.

The remainder of the paper is structured as follows. Section 2 briefly reviews related work on computational models of technology and industry evolution. Section 3 presents the framework. Section 4 describes our model of platform evolution and our results to date. Section 5 concludes.

2. RELATED WORK

Prior Models of Technology Evolution

There is a long tradition of modeling technological change as a search process over randomly selected production possibilities. Nelson and Winter [32] provide an early example; more recent work has employed variations on the *NK* model [6, 25]. These models exhibit technological progress (often uneven) arising out of investments in research and development, but for the most part they abstract away from the structure of the technologies themselves.

In contrast, Arthur and Polak [5] developed a model that is grounded in the domain of electronic logic circuits. This approach allowed them to postulate an empirically plausible fitness function, since the properties of any valid combination of logic gates are well-defined and can be compared with

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.

known circuits that perform useful functions, like adding numbers. (While their model was not intended to invent circuits that are novel as well as useful, others have actually done so—notably Koza et al. [27, 28].)

A third stream of work has modeled economic production as a set of linked processes akin to a complex chemical reaction [33], based on the work of Eigen and Schuster [14]. This work yields insight into the emergence of novel self-reinforcing loops (hypercycles), but is difficult to adapt to study the evolution of designed artifacts.

Altenberg's Generalized NK Model and Frenken's Mapping to Technology Landscapes

Mindful of the power of real-world fitness functions, the pitfalls of synthetic ones, and the promise of alternative approaches, we take the familiar *NK* model as our starting point. Despite its stark assumption of "total ignorance" about the interactions among the elements of a complex system [22], we believe the *NK* model can be adapted to model technology evolution in a fruitful way.

An important but frequently overlooked generalization of the *NK* family was proposed by Altenberg [2, 3] and first applied to technological innovation in complex engineered systems by Frenken [18]. This "generalized *NK* model" distinguishes structural elements (genes or system components) from functional attributes (traits or product features), and allows arbitrary patterns of interdependence between them. Relaxing the requirement that each element interact epistatically with (i.e., modify the effects of) exactly K others yields genotype–phenotype maps in which elements can vary in their pleiotropy (the number of functions with which they interact) and functions can vary in their polygeny (the number of elements that affect them).

Altenberg used this model to explore the growth of biological genomes through "constructional selection," an evolutionary process in which new genes are added when they improve an organism's total fitness. He found that genes with high pleiotropy tend to be added early, while later additions typically affect only a few functional attributes. Constructional selection tends to increase fitness by rejecting new genes that interfere with parts of the genome that are already well adapted, while accepting those that yield incremental improvements or deliver new functions.

Frenken recognized that analogous processes occur in technological evolution, and employed Altenberg's model to explain well-known patterns in product and technology life cycles [1, 13]. He defined a technological paradigm as "a set of standardised high-pleiotropy elements in the product population," and interpreted Altenberg's results on constructional selection to show how "an invariant core of high-pleiotropy elements emerges endogenously in a growing complex system" (p. 62). He further observed that many types of design changes (including architectural, modular, incremental, and radical innovation [20]) can be expressed using generalized genotype–phenotype maps, which makes such maps a potentially powerful tool for studying evolution on complex technology landscapes.

However, the Altenberg–Frenken constructional selection model lacks two ingredients needed to build a fully dynamic model of technology evolution. First, the product population is not explicitly modeled. Each simulation run consists of a single individual (genome or system) that evolves in isolation. This precludes studying competition between alternative designs that differ in their complexity or specific arrangement of components. Second, new genes or components can only be added, never removed or substituted for variants. This assumption may be defensible in a biological setting (where genomes tend to grow through accretion, even if many parts are deactivated by regulatory mechanisms), but it fails to capture the repertoire of actions available to human designers. Fortunately, both ingredients are readily available in the modeling literature.

Levinthal's Treatment of Adaptation and Population Selection

In a seminal paper that introduced *NK* modeling to management scholars [29], Levinthal studied adaptive behavior in a population of agents on a complex fitness landscape. Agents representing organizations engage in both local search and "long-jump" adaptation (testing random organizational forms against the current one). In addition, the model includes a population-level process of change based on assumptions derived from the biological literature:

- In each period, the probability that an organization survives is proportional to the ratio of its fitness to that of the most fit organization in the population. Nonsurvivors are replaced in order to maintain a constant population size.

- A new organization may either replicate an existing one or choose a configuration of attributes at random. The probability of choosing attributes at random is proportional to the genetic load of the population (a concept borrowed from population biology [36]), which is defined as the one minus the ratio of the average fitness value to the maximum fitness value.

- If a new organization is chosen to replicate an existing one, the probability of a given organization being selected as a "target" is determined by its relative fitness in the population, with more fit forms being proportionately more likely to be replicated.

These assumptions are orthogonal to the key assumptions of the Altenberg–Frenken model, in the sense that fitness landscapes based on generalized genotype–phenotype maps can be substituted for the traditional *NK* landscapes studied by Levinthal (although to our knowledge we are the first to do so). This is the essence of the framework presented below.

3. A FRAMEWORK FOR MODELING TECHNOLOGY EVOLUTION ON RANDOM LANDSCAPES

This section presents a framework for modeling the evolution of technologies in a modern economy. As in the Altenberg–Frenken model, the main elements of the framework are system components and their associated functions.[1] Figure 1 illustrates the relationships among these elements.

[1]To build intuition, we use Frenken's technological vocabulary (components, functions, systems) in the context of technology evolution, instead of Altenberg's biological terms (genes, traits, organisms) or the organizational ones (decisions, departments, firms) typically used by management scholars to in the context of *NK* models.

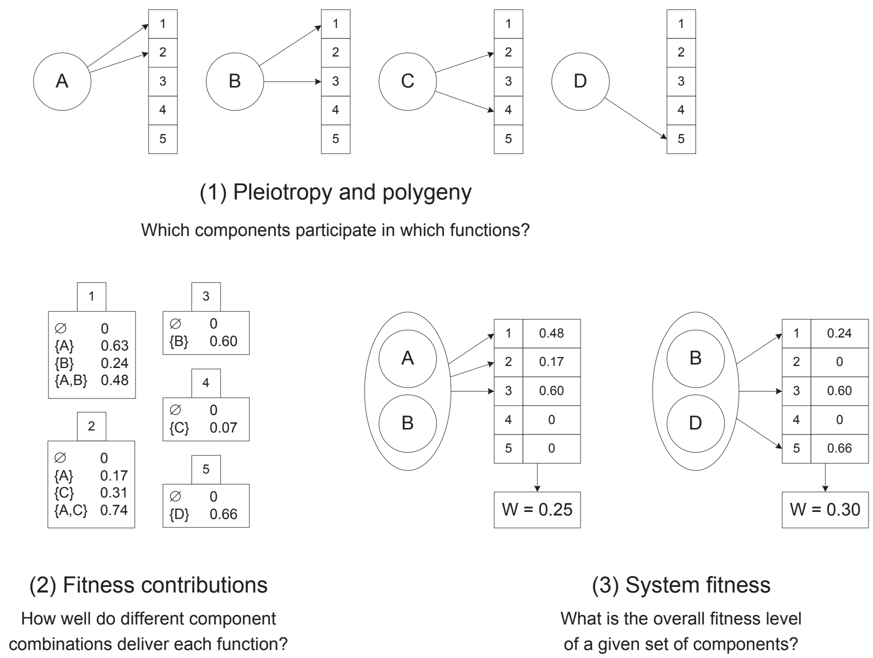

Figure 1: Relationships among components, functions, and systems.

Pleiotropy and Polygeny

Each component (A, B, C, D) enables (or affords [31]) a set of functions $(1, 2, 3, 4, 5)$. These relationships are indicated by arrows in the figure. When multiple components enable the same function (polygeny), the effects of one component may be modified by the presence or absence of others (epistasis). Likewise, a component may enable multiple functions (pleiotropy). Adapting Altenberg's terminology [2, 3], we call the set of functions that a given component enables its *pleiotropy set*, and the set of components that enable a given function its *polygeny set*.

Fitness Contributions

Different combinations of components interact differently in enabling their respective functions; some of these interactions may be positive, while others may be negative. Each function is associated with a *fitness contribution* whose value depends on the presence or absence of the components in the function's polygeny set. The effects of these dependencies are shown in the figure by the tables associated with each numbered function. For example, function 1 contributes a value of 0.63 if component A is present but B is not, but only 0.48 if both A and B are present. (Note that B contributes separately to function 3, which compensates for its negative effect on function 1.)

Following standard practice in the *NK* literature, we adopt the baseline assumption that fitness contributions are uncorrelated with each other. In other words, adding or removing a component yields a new random value that is independent of the previous one. (We use correlated fitness landscapes to model heterogeneous consumer preferences in the model of platform evolution discussed below.) Also following standard practice, fitness contributions are denoted by real numbers distributed uniformly on the unit interval.

Unlike both traditional *NK* models and the generalized Altenberg–Frenken model, our framework does not explicitly distinguish the effects of alternative component designs (alleles). This approach simplifies models that focus on the selection of components from a large and/or changing population, in contrast to the usual focus on the search for high-value configurations within a fixed design space.[2]

Design Fitness (Value Creation)

A system, for our purposes, is simply a set of components. A system design can thus be represented as a bit string in which each position corresponds to a component, and a 1 denotes the presence of that component while a 0 denotes its absence.

When a system is deployed in a particular environment, it creates economic value by satisfying the needs of its users. For simplicity, we associate every system with an overall fitness level, labeled *design fitness*, that depends only on the identity of its components (the system's "genotype"). Design fitness expresses the value of the functionality delivered by the system (the "phenotype"); it can be interpreted as a normalized measure of users' aggregate willingness to pay for systems that realize the design.

Again following standard practice, a system's design fitness is defined as the mean of the functions' fitness contributions. We need an additional assumption to define the fitness contribution of a function that is not enabled by any component in the system (a situation that does not arise in the standard *NK* model). We follow Altenberg [2, 3] in assuming that such functions have a contribution of zero.

[2]This simplification is without loss of generality, since models are free to define the set of available components in many ways. An easy way to restore the structure of a traditional *NK* model is to assume that components come in pairs $(A, \bar{A}; B, \bar{B}; \ldots)$ corresponding to binary alleles ("on" vs. "off" states). Then a one-mutation adaptive walk would be defined as a swap of one component for its allelic sibling.

Designer Fitness (Value Capture)

Depending on the structure of the environment (e.g., the number of consumers and firms, as well as their respective preferences and pricing policies), some fraction of a system's value is captured by its designers in the form of economic rent or profit, which in turn may influence the future value of the system (e.g., through reinvestment in product development). Recognizing that technology evolution operates at multiple levels, from the cognitive processes of individual designers to the market forces that shape the population of organizations that employ them, we distinguish the fitness of designers from the fitness of their designs.

The assumptions relating design fitness to *designer fitness* necessarily vary across models, just as assumptions about the relationship between value creation and value capture vary across traditional economic models. To accommodate this variation, we leave these assumptions to be specified by each model rather than the framework itself.

For example, in the platform evolution model presented below, the full value of a system design may be credited to multiple designers. This represents an ideal case in which the designers' incentives are aligned to maximize the value of the system as a whole. In reality, however, incentive conflicts can occur because designers typically compete for shares of a finite economic "pie" denominated in monetary units. Our most recent work (not reported in this paper) grapples with this constraint more directly.

Evolutionary Dynamics

In order to integrate the elements of the framework into a dynamic model of technology evolution, an evolutionary process must be specified [11, 21]. This process does not need to capture the full complexity of competition, investment, market share, and survival. But it does need to generate sufficient variety among designs, and selective pressure among designers, to effectively explore the vast combinatorial design spaces that can easily arise in such models.[3] In the platform model described in the next section, we focus on processes that include both myopic search and population selection. These processes favor the survival of higher-value systems and their associated designers.

4. APPLICATION TO THE EVOLUTION OF PLATFORM ARCHITECTURES

We are currently applying the framework presented above to study the evolution of platform architectures using an agent-based model. Baldwin and Woodard [8] define a platform as "a set of stable components that supports variety and evolvability in a system by constraining the linkages among the other components." They propose that platforms arise due to the interaction of two fundamental economic forces: pressure to reuse certain "core" components (e.g., due to economies of scale in production or the difficulty of finding substitutes) while facilitating variation in other "peripheral" components (e.g., to satisfy diverse consumer preferences or increase resilience to environmental change). The initial aim of our study is to replicate these forces *in silico* and explore the conditions under which platform architectures emerge. More broadly, we seek to develop a robust model of platform evolution that can be used to study strategic design choices by participants in a platform ecosystem.

This work is complementary to the existing literature on platform competition [15, 16, 19], which typically assumes the existence of one or more platforms and then studies economic phenomena such as pricing and consumer welfare under game-theoretic assumptions of equilibrium behavior. In our model, by contrast, platform architectures can emerge endogenously and the network structure of component reuse can be observed directly, both in cross section and over time.

Basic Model

Consistent with our framework, the main elements of the model are *components* that enable various *functions* and can be combined into *systems* whose value to their users depends only on the combination of components they contain.

Inspired by the metaphor of the "primordial soup" in which biologists sometimes imagine life to have arisen, we assume an unlimited supply of components that come in C discrete types (like inorganic "feedstock" compounds). A system can perform up to F functions. The set of functions enabled by a particular component is fixed at the start of a simulation run (with each component being "wired" to a given function with a fixed probability), and remains unchanged for the duration of the run. The value created by a particular system (i.e., its design fitness) is defined as in section 3, with individual fitness values drawn uniformly at random on the unit interval.

To explore the emergence of stable sets of components (i.e., potential platforms), we introduce an intermediate level of aggregation between components and systems. Firms, which we call *producers*, assemble components into *products* that are in turn purchased by *consumers*, who can either use them alone or assemble multiple products into systems. For simplicity, we assume that product boundaries are irrelevant in evaluating design fitness; consumers assign the same value to a given set of components (e.g., $\{A, B, C\}$) whether they are purchased as a single standalone product or assembled separately (e.g., $\{A, B\}$ and $\{C\}$). Duplicate instances of a component are also assumed to be irrelevant (e.g., combining $\{A, B\}$ and $\{B, C\}$ is equivalent to simply purchasing $\{A, B, C\}$).

In a given period, there are N producers. Each producer has a "strategy" that consists of a single product design (i.e., combination of components). Producers' strategies need not be unique; many producers can make the same design. Each producer makes one unit of its design in each period. After production occurs, consumers assemble systems from the products available in the market. After all products have been consumed, population selection occurs among producers, simulating the action of competitive markets. Finally, the cycle repeats until T periods have elapsed.

Below we elaborate on the central drivers of the model's evolutionary dynamics: myopic search by consumers and population selection among producers.

Myopic Search by Consumers

After production occurs, there is a pool of N products available for consumers. Consumers arrive one at a time. Each

[3] Even under the assumption that systems are defined only by the presence or absence of components (i.e., ignoring all details about their configuration and assembly), the size of the design space scales by the familiar powers of two: 16 components yield 65,535 distinct system designs, 32 components yield over 4 billion designs, and 64 components yield over 10 quintillion.

consumer then assembles a system through a greedy myopic gradient search process, as follows:

- The consumer examines the pool and selects, uniformly at random, G products to evaluate (or all of the remaining products if fewer than G are available).
- To evaluate a product, the consumer adds it to his/her current system, computes the value of the new system as prescribed above, and compares it to the value of the current system.
- If at least one of the products evaluated by the consumer yields a system with strictly higher value than the current one, the product that yields the greatest improvement is kept and the search process begins again with a newly selected set of products (if any remain in the pool). If not, this consumer exits the market and a new consumer enters.

Consumers continue to arrive and assemble systems until the pool of available products is exhausted.

Population Selection Among Producers

At the end of each period, population selection occurs among producers. To operationalize the concept of designer fitness as defined in section 3, we define the fitness of a producer as the value of the system containing the product it produced in the current period. This assumption rewards producers whose products contribute to high-value systems.[4]

The remaining assumptions on population selection closely follow Levinthal [29]:

- Each producer survives with probability proportional to the ratio of its fitness to the maximum fitness in the population.
- Producers that do not survive are replaced by new entrants. A new entrant is either a raw component supplier (i.e., its product consists of a single component, chosen uniformly at random from the set of component types) or an imitator of an existing producer.
- The probability of imitating an existing producer is inversely proportional to the genetic load of the producer population [36].

An entrant assigned to imitate an existing producer may also innovate by "mutating" the product design of the producer it is imitating. With probability p, an imitating firm does each of the following before finalizing its design:

- Add a component chosen uniformly at random from the available types.
- Drop a component chosen uniformly at random from its current product design.
- Swap two components by performing an add followed by a drop.

[4]There is also an element of chance, since a producer cannot directly control which consumer buys its product in a given period. If G is low (or the consumer's search process gets "stuck" at a local optimum [34]), it is possible for a product to be used in a system to which it contributes less value than it would in alternative uses. In other words, we do not assume an efficient market.

Extensions

As noted earlier, our modeling effort was motivated by the proposition that platform architectures arise in response to economic forces. These forces are present to a limited extent in the basic model. Sets of components with high fitness contributions that interact positively (or at least not too negatively) with a variety of other components are likely to exist but be difficult to find, hence firms that package these components together as a product—by sheer luck, since we do not model strategic behavior by producers—are likely to enjoy a selective advantage, increasing both their longevity and the rate at which their designs are imitated by new entrants. So we should not be surprised to find some degree of platform emergence even in the basic model.

That said, the basic model omits three forces that we would expect to drive the emergence of platforms even more strongly. One is economies of scale, which we do not address in the present version of the model. The other two are environmental change and consumer heterogeneity. We implemented both of these as extensions to the basic model.

Environmental Change

Environmental change can be interpreted as change in consumer preferences or in the ways that products interact with their environment. If environmental change is enabled in a given simulation run, it occurs in each period with probability c, called the frequency of change. If environmental change occurs, each function is "scrambled" with a probability s, called the severity of change. If a function is selected to be scrambled, all of the fitness contributions associated with that function are randomly redrawn. This is similar to the approach of Siggelkow and Rivkin [35].

Allowing the frequency and severity of change to be varied independently allows us to investigate very different types of change, from rare but catastrophic disruptions (like the shift from mainframes to personal computers) to those that are more frequent but moderate in scope (like the substitution of fuel injection systems for carburetors).

Consumer Heterogeneity

When consumers' preferences are diverse, there is almost certainly no single design that is viewed as "best" by all of them. We model this phenomenon by dividing consumers into L groups uniformly at random, and giving each group its own fitness landscape. (Recall that in our context, a fitness landscape is a mapping from system designs, i.e., sets of components, to real numbers representing value. In the basic model we implicitly assumed that all consumers value systems the same way, yielding a single fitness landscape.)

The fitness values on these landscapes may be correlated, with the extent of correlation given by the parameter r. If $r = 0$, each landscape is uncorrelated with the others; in other words, their fitness contributions are drawn independently and identically from the same distribution. More generally (if $L > 1$ and $r > 0$), we define a "parent" landscape and L "child" landscapes. For each child landscape, each function is either inherited from the parent or unique to the child. In order to ensure the same degree of correlation across all child landscapes, we want the same functions to be inherited or unique for each; we arbitrarily set the first $F \cdot r$ functions to be inherited, and the remaining $F\,(1-r)$ to be unique.

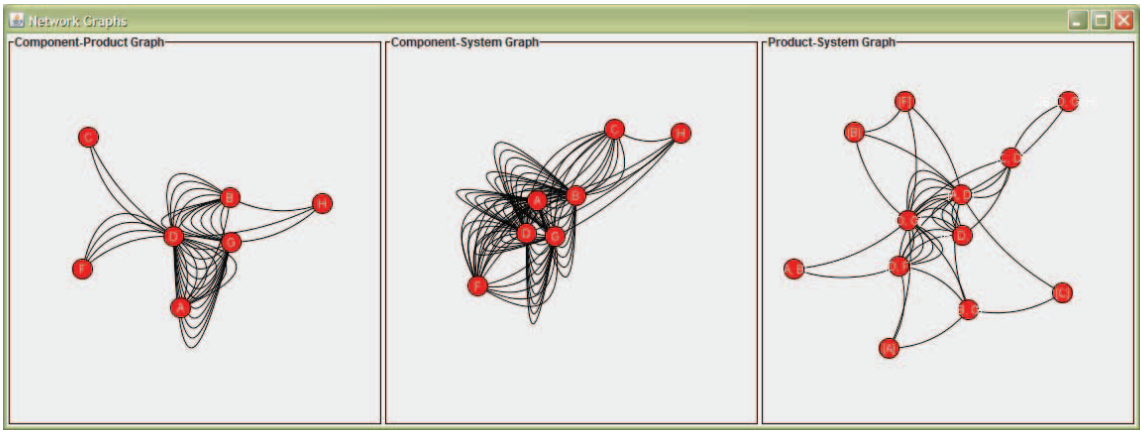

Figure 2: Screen shot of an interactive run. Network ties in the three panels respectively represent co-occurrence of components in products, components in systems, and products in systems.

Experiments

We implemented the model in Java using the MASON simulation toolkit [30], and are conducting experiments on a Linux-based high-performance computing cluster.[5] All of our reported results are averaged over 100 independent trials for each combination of parameters, with a different random number seed for each trial. All observations are taken after 200 time periods; we verified the robustness of our results to run lengths ranging from 100 to 500 periods.

Figure 2 shows a screen shot from an interactive run of the model with 8 components and 30 producers. The graph on the left illustrates the co-occurrence of components in products (e.g., the two edges between C and D indicate that two products in the population included these components during the time period when the screenshot was taken). Similarly, the middle graph illustrates the co-occurrence of components in systems (e.g., the single edge between C and H indicates that these two components were present in exactly one system; since these components are not linked in the component–product graph, we can infer that they were brought together during system assembly by a consumer). The graph on the right illustrates the co-occurrence of products and systems; here the nodes are products rather than individual components.

We ran a set of preliminary simulations that explored the parameters of the model, including number of components (C), number of functions (F), and the fraction of products evaluated by each consumer at each step in the system assembly process (G/N). While there were some interesting differences (which we will discuss in an expanded version of this paper), the results presented below are qualitatively robust to a wide range of parameter values.

Average Product Centrality

In addition to a variety of other outcome variables (e.g., product size, system size, and system value), we constructed a measure of platform emergence called *average product centrality*. This measure is motivated by our desire to identify not just fully formed and easily recognizable platforms, but also proto-platforms: configurations of products and systems with platform-like characteristics. Intuitively, average product centrality measures the extent to which the product–system graph exhibits the kind of pattern shown in the right-hand panel of Figure 2, where some products form a densely connected core surrounded by a sparsely connected periphery. These core products are platform-like in the sense that they are frequently reused (i.e., they appear in many different systems) and they support variety (i.e., they appear in combination with a wide range of other products).

To compute the average product centrality, a centrality score is computed for each product and then averaged over the product population. Since a product is a set of components, we computed the centrality score using a network measure designed for groups rather than individuals, namely a weighted and normalized variant of the *group betweenness centrality* measure proposed by Everett and Borgatti [17]. Group betweenness centrality (GBC) is defined as the proportion of geodesics connecting pairs of non-group members that pass through the group—in other words, the extent to which components appear in systems more often in combination with a particular product than with others.

Formally, let P be a product. Let $g_{u,v}(P)$ denote the number of geodesics (distinct shortest paths) in the component–system graph that connect components u and v without passing through components in P. Let $g_{u,v}$ denote the total number of geodesics connecting u and v, and define an ordering ($<$) on the set of components. Then:

$$\mathrm{GBC}(P) = \sum_{u<v} \frac{g_{u,v}(P)}{g_{u,v}} \text{ for } u, v \notin P.$$

Key Results

Figures 3 and 4 summarize our key results to date. The vertical axis in both graphs is the average product centrality measure defined above, and both graphs are shaded using the same color scheme.

Reasoning by analogy from biological evolution, we expected a complex and non-linear relationship between frequency and severity of change and the emergence of platforms. Environments that are very stable lead to stable,

[5]We also used several other free and open-source Java libraries (e.g., JUNG for network visualization, Guice for injecting parameter values into model classes, and Hibernate for storing observations to a database during simulations).

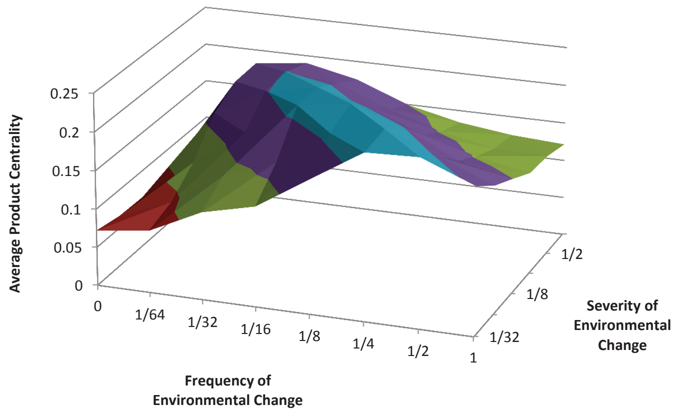

Figure 3: Platform emergence varies with frequency and severity of change.

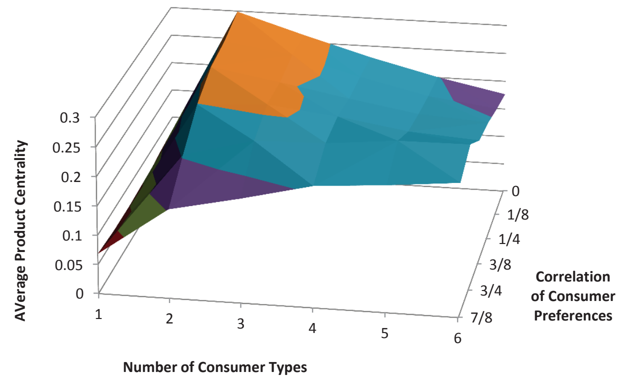

Figure 4: Platform emergence varies with the number of niches and the correlation between them.

optimized systems that are often produced by a single integrated firm. As environmental change increases, it becomes advantageous to produce modules that can be reassembled in a variety of ways by a variety of producers [7]. In the technologically more stable markets of the 1960s, IBM was a vertically integrated computer company, GM was a vertically integrated car company, and AT&T was a vertically integrated telecommunications company. In the more turbulent environment of the early 21st century, it is more profitable to produce disk drives, or processor chips, or office software, that can be "mixed and matched" to satisfy changing market demand. When environmental change becomes too rapid or too severe, however, evolution can no longer select for products that work well together, and even well-adapted products fall victim to random shocks.

Figure 3 exhibits a striking visual pattern that supports this prediction: a diagonal "ridge line" that peaks in the region of frequent but moderate change. This figure shows that platform architectures are most strongly favored under precisely the conditions in which it is most valuable to have a stable set of core components that work well with a variety of peripheral ones. Most importantly for our research, we were able to induce the emergence of platforms without explicitly modeling participation externalities favoring widespread adoption or economies of scale favoring reuse, and without memory, learning, or anticipation on the part of product designers.

Introducing consumer heterogeneity in the form of multiple market niches yielded results that were qualitatively similar to environmental change, as shown in Figure 4. As the number of market niches increases, so does the advantage of having a product that can be combined into systems with high fitness in multiple niches. Each niche represents a subpopulation of consumers whose ideal products differ in one or more functional attributes (e.g., business or home computer users). As niches diverge, they begin to resemble separate, unrelated markets, and it is difficult to develop products that can succeed in all of them. Not surprisingly, the maximum average product centrality appeared under consumer heterogeneity rather than under environmental change, because in a stable environment evolution has more time to converge to a population of well-adapted products.

5. CONCLUSION

In this paper, we presented an approach to modeling technology evolution on random fitness landscapes. This approach builds on prior work that extended Kauffman's *NK* model using the idea of a generalized genotype–phenotype map, and extends that work in turn by simulating the evolutionary dynamics among product and system designers using agent-based modeling techniques. Our preliminary efforts to study the evolution of platform architectures using these ideas have yielded promising initial results. We plan to explore a set of related issues using the same tools. More broadly, we believe the framework presented in the paper can be used to illuminate a variety of phenomena related to the evolution of complex engineered artifacts and systems.

6. ACKNOWLEDGMENTS

Both authors thank the School of Information Systems at Singapore Management University and the Wharton–SMU Research Centre for generous financial support.

7. REFERENCES

[1] W. J. Abernathy and J. M. Utterback. Patterns of industrial innovation. *Technology Review*, 80(7):40–47, 1978.

[2] L. Altenberg. Evolving better representations through selective genome growth. In *Proceedings of the IEEE World Congress on Computational Intelligence, Part 1*, pages 182–187, 1994.

[3] L. Altenberg. Genome growth and the evolution of the genotype-phenotype map. In W. Banzhaf and F. H. Eckman, editors, *Evolution and Biocomputation*, pages 205–259. Springer-Verlag, 1995.

[4] W. B. Arthur. *The Nature of Technology: What It Is and How It Evolves*. Free Press, 2009.

[5] W. B. Arthur and W. Polak. The evolution of technology within a simple computer model. *Complexity*, 11(5):23–31, 2006.

[6] P. Auerswald, S. Kauffman, J. Lobo, and K. Shell. The production recipes approach to modeling technological innovation: An application to learning

by doing. *Journal of Economic Dynamics and Control*, 24(3):389–450, 2000.

[7] C. Y. Baldwin and K. B. Clark. *Design Rules, Vol. 1: The Power of Modularity*. MIT Press, 2000.

[8] C. Y. Baldwin and C. J. Woodard. The architecture of platforms: A unified view. In A. Gawer, editor, *Platforms, Markets and Innovation*, pages 19–44. Edward Elgar, 2009.

[9] E. D. Beinhocker. *The Origin of Wealth: Evolution, Complexity, and the Radical Remaking of Economics*. Harvard Business School Press, 2006.

[10] E. D. Beinhocker. Evolution as computation: Integrating self-organisation with generalized darwinism. *Journal of Institutional Economics*, 7(3):393–423, 2011.

[11] D. T. Campbell. Variation and selective retention in socio-cultural evolution. *General Systems*, 14:69–85, 1969.

[12] K. A. De Jong. *Evolutionary Computation: A Unified Approach*. MIT Press, 2006.

[13] G. Dosi. Technological paradigms and technological trajectories: A suggested interpretation of the determinants and directions of technical change. *Research Policy*, 11:147–162, 1982.

[14] M. Eigen and P. Schuster. *The Hypercycle: A Principle of Natural Self-organization*. Springer-Verlag, 1979.

[15] T. Eisenmann, G. Parker, and M. Van Alstyne. Platform envelopment. *Strategic Management Journal*, 32(12):1270–1285, 2011.

[16] D. S. Evans, A. Hagiu, and R. Schmalensee. *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*. MIT Press, 2006.

[17] M. G. Everett and S. P. Borgatti. The centrality of groups and classes. *Journal of Mathematical Sociology*, 23(3):181–201, 1999.

[18] K. Frenken. *Innovation, Evolution and Complexity Theory*. Edward Elgar, 2006.

[19] A. Gawer and M. A. Cusumano. *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*. Harvard Business School Press, 2002.

[20] R. M. Henderson and K. B. Clark. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly*, 35(1):9–30, 1990.

[21] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.

[22] S. A. Kauffman. *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press, 1993.

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

[24] S. A. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128(1):11–45, 1987.

[25] S. A. Kauffman, J. Lobo, and W. G. Macready. Optimal search on a technology landscape. *Journal of Economic Behavior and Organization*, 43(2):141–166, 2000.

[26] S. A. Kauffman and E. D. Weinberger. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology*, 141(2):211–245, 1989.

[27] J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane. *Genetic Programming III: Darwinian Invention and Problem Solving*. Morgan Kaufmann, 1999.

[28] J. R. Koza, M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu, and G. Lanza. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer, 2003.

[29] D. A. Levinthal. Adaptation on rugged landscapes. *Management Science*, 43(7):934–950, 1997.

[30] S. Luke, C. Cioffi-Revilla, L. Panait, and K. Sullivan. MASON: A new multi-agent simulation toolkit. In *Proceedings of the Eighth Annual Swarm Users/Researchers Conference (SwarmFest 2004)*, 2004.

[31] J. R. A. Maier and G. M. Fadel. Affordance based design: A relational theory for design. *Research in Engineering Design*, 20(1):13–27, 2009.

[32] R. R. Nelson and S. G. Winter. *An Evolutionary Theory of Economic Change*. Harvard University Press, 1982.

[33] J. F. Padgett, D. Lee, and N. Collier. Economic production as chemistry. *Industrial and Corporate Change*, 12(4):843–877, 2003.

[34] J. W. Rivkin and N. Siggelkow. Organizational sticking points on NK landscapes. *Complexity*, 7(5):31–43, 2002.

[35] N. Siggelkow and J. W. Rivkin. Speed and search: Designing organizations for turbulence and complexity. *Organization Science*, 16(2):101–122, 2005.

[36] E. O. Wilson and W. H. Bossert. *A Primer of Population Biology*. Sinauer Associates, 1971.