

# **The Evolution of Modular Product Architectures and the Emergence of Platform Ecosystems**

**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

16 January 2013

## **ABSTRACT**

While a large body of research has examined the advantages and disadvantages of modularity in product design, studies of modularization have tended to focus on settings in which a product's architecture is determined by a single designer or designers within a single organization. In this paper, we ask whether modularity can arise in a decentralized setting where high-value designs are rewarded but designers' ability to coordinate with each other and anticipate the consequences of their actions is limited—in short, whether modularity can evolve. To answer this question, we developed an agent-based model in which boundedly rational firms combine components into products. We conducted computational experiments to identify conditions that favor the emergence of products with a high degree of modularity, as measured by the extent of functional decoupling among the components. We further explored the extent to which modularity enables firms to discover high-value product designs in environments characterized by diverse or dynamic consumer preferences. Finally, we investigated the patterns of reuse among components to trace the emergence of “core” products that can become the basis for platform ecosystems. Our preliminary results support the conclusion that modularity in product design can indeed evolve, although the products generated by our model lack the defining features of real-world modular architectures (e.g., hierarchically nested subsystems and interface standards). Nonetheless, these findings can serve as a baseline for future research on the evolution of complex products and the industries that produce them.

## **Keywords:**

Modularity; product architecture; simulation

# 1 Introduction

Modularity is an essential strategy for managing the complexity of modern products and services, from computers to aircraft engines to financial services (Baldwin & Clark, 1997; Brusoni & Prencipe, 2001; Jacobides 2005). Among other properties, modular architectures offer the ability to split up tasks among designers, make changes to one part of a system while minimizing the effects on others, and stimulate innovation by an “ecosystem” of competitors and complementors (Simon, 1962; Ulrich, 1995; Baldwin & Clark, 2000). While a large body of research has examined the advantages and disadvantages of modularity in product design, studies of modularization have tended to focus on settings in which a product’s architecture is determined by a single designer or designers within a single organization.

In this paper, we ask whether modularity can arise in a decentralized setting where high-value designs are rewarded but designers’ ability to coordinate with each other and anticipate the consequences of their actions is limited—in short, whether modularity can evolve. To answer this question, we developed an agent-based model in which boundedly rational firms combine components into products. Our model is based on a generalization of the NK family of fitness landscape models (Kauffman & Levin, 1987; Kauffman & Weinberger, 1989), which we adapted to study populations of product designs in which different products have different architectures. This distinctive feature of our approach allows us to study product architecture as the endogenous outcome of an evolutionary process.

We conducted computational experiments to identify conditions that favor the emergence of products with a high degree of modularity. To measure modularity, we adapted the measure introduced by Frenken (2006), which allows the extent of functional decoupling within products of different size and scope to be compared in a meaningful way. Frenken and Mendritzki (2012) argue that modularity can speed up problem-solving in settings where the set of design choices cannot be decomposed into non-overlapping subsets. This insight motivated us to speculate that when product architecture is endogenous, modularity should be

more prevalent in settings where finding high-value designs is more difficult (e.g., environments characterized by diverse or dynamic consumer preferences). In addition to seeing more modularity, we should also find a positive correlation between modularity and value—not because modular designs are inherently better than non-modular ones, but because high-value designs are simply easier to find if they are modular than otherwise.

Our preliminary results support the conclusion that modular product architectures can indeed evolve. We find that the average level of modularity at the end of our simulations is significantly higher than we would expect from a collection of randomly selected components, and that modularity is associated with higher-value products (after controlling for other factors). We also investigated the patterns of reuse among components to trace the emergence of “core” products that can become the basis for platform ecosystems. We found, as expected, that such products were more prevalent in settings where search was more difficult.

Despite these promising findings, our results should be seen as preliminary. We failed to find evidence for some of the key causal mechanisms we anticipated, such as the role of subsystem suppliers in catalyzing the emergence of modularity. Even effects that were consistent with our expectations, such as the link between modularity and value, were in some cases too small to be visible without resorting to regression techniques. Moreover, most of the products generated by our model lack the defining features of real-world modular architectures (e.g., hierarchically nested subsystems and interface standards). These findings can nonetheless serve as a baseline for future research on the evolution of complex products and the industries that produce them.

The rest of the paper is organized as follows. Section 2 briefly reviews the relevant literature on NK modeling, modularity, and technology evolution. Section 3 presents our model and the extensions we implemented. Section 4 describes our computational experiments and results to date. Section 5 concludes by discussing the significance and limitations of the results, as well as opportunities for future work.

## 2 Related Literature

We build heavily on the generalized NK model of Altenberg (1994), which was adapted to the domain of complex technological systems by Frenken (2006). Other scholars who have used the NK framework to study modular design include Marengo et al. (2000), Ethiraj and Levinthal (2004), Dosi and Marengo (2005), Brusoni et al. (2007), Rivkin and Siggelkow (2007), and Frenken and Mendritzki (2012). Our model is also indebted to Levinthal (1997), whose approach to population selection on NK landscapes we follow closely.

More broadly, we aim to contribute to the growing literature on technology evolution (Arthur, 2009) and the economics of complex adaptive systems (Beinhocker, 2006). While the present model does not allow us to directly address questions concerning platform competition and strategy (Gawer & Cusumano, 2002; Iansiti & Levien, 2004; Evans et al., 2006), our hope is that it could be extended to support these kinds of questions in the future.

## 3 Model

The main elements of the model are designs and designers, which for concreteness we call products and firms. A product is a set of components, each of which enables one or more functions. Each function, in turn, contributes value to the product as a whole. More than one component can enable the same function in a given product, in which case the components interact. These interactions can be positive or negative. Product design simply entails choosing a set of components; we do not explicitly consider the details of their configuration or assembly. Also for simplicity, each firm is limited to a single product.

Firms engage in product design repeatedly over a series of time periods. In each period, each firm is presented with an opportunity to modify its current design by adding or removing components. These changes can be viewed as movements on a high-dimensional fitness landscape. The value of a firm's product (i.e., its "height" on the landscape) affects the likelihood that the firm survives to the next period. Firms that do not survive are replaced by new

entrants, each of which either replicates the design of a surviving firm or starts from scratch with a new product. The total population of firms is held fixed by setting the number of entrants in a given period to the number of non-survivors from the previous period.

In our experiments, we studied three extensions to this basic model. First, we allowed firms to modify their designs by adding or removing not just individual components but entire products of other firms. Second, we modeled environmental change (e.g., in technology or consumer preferences) by periodically randomizing the fitness landscape for product designs. Third, we modeled the existence of multiple market segments by creating multiple fitness landscapes and associating different firms with different landscapes. The remainder of this section specifies the basic model more formally and describes the extensions in greater detail.

### 3.1 A Rugged Fitness Landscape for Designs

Let  $C$  denote the number of available components and  $F$  denote the number of functions they collectively enable. Following Altenberg (1994), the number of functions enabled by a given component is called its *pleiotropy*, and the number of components that enable a given function is called its *polygeny*. The interactions between components and functions can be represented as a matrix with components on the rows, functions on the columns, and either 0 or 1 in each cell to indicate whether a given component enables the respective function. We call this matrix the *component–function map*; it is also known as a genotype–phenotype map. We call the set of functions enabled by a given component its *pleiotropy set*, and the set of components that enable a given function its *polygeny set*.

A product design is completely specified by the set of components it contains. Thus there are  $2^C$  possible product designs (including an “empty” design that contains no components), each of which can be represented as a binary string of length  $C$ , with a 0 or 1 in position  $i \in \{1 \dots C\}$  indicating the absence or presence of the  $i$ th component. Using the component–function map, it can easily be determined which functions are enabled by a given product and which combinations of components enable each function, as illustrated in Figure 1.

(In product {A, B}, function 1 is enabled by both components, function 2 is enabled only by component A, and function 3 is enabled by component B. In product {B, D}, functions 1 and 3 are enabled only by component B, and function 5 is enabled by component D.) We call the set of components that enable a given function in a particular product the *enabling set* of that function.

To determine the value (or fitness)<sup>1</sup> of a product  $\mathbf{p}$ , the product is evaluated with respect to each function  $j \in \{1 \dots F\}$  to determine the level of functionality (or quality) that the product delivers for this function. This yields a vector of *function values*, each of which depends only on the presence or absence of components in the corresponding function's polygeny set. Let  $\mathbf{g}_j$  denote the polygeny set of function  $j$  and  $\mathbf{e}_j(\mathbf{p}) = \mathbf{p} \cap \mathbf{g}_j$  denote the enabling set of product  $\mathbf{p}$  with respect to this function. We randomly assign a function value  $v_j(\mathbf{e}_j(\mathbf{p})) \in [0,1]$  to each of the  $2^{|\mathbf{g}_j|}$  possible non-empty values of  $\mathbf{e}_j(\mathbf{p})$  by drawing independently from a uniform distribution. This assumption is akin to “confess[ing] our total ignorance and admit[ting] that ... essentially arbitrary interactions are possible” among the components that enable a given function (Kauffman, 1993, p. 41). The *product value* is then defined as the average of the  $F$  function values:  $V(\mathbf{p}) = \frac{1}{F} \sum_{j=1}^F v_j(\mathbf{e}_j(\mathbf{p}))$ .

So far these assumptions are consistent with the generalized NK model of Altenberg (1994) and Frenken (2006). In addition, we assume that if a function is not enabled by any components in a given product (i.e.,  $\mathbf{e}_j(\mathbf{p}) = \emptyset$ ), then its value is zero.<sup>2</sup> This introduces a fundamental asymmetry into the model, namely that an absent component cannot create any value. In a standard NK model, by contrast, there is no special distinction between “0” and “1” bits of a genotype, which are typically interpreted as alternative choices on a given dimension (“alleles”).<sup>3</sup> This assumption will play a key role in allowing us to study the evolution of modular

---

<sup>1</sup> In this paper, we use the terms “value” and “fitness” interchangeably. We do not claim that the underlying concepts are equivalent, as each raises subtle issues of interpretation (e.g., value to whom, fitness with respect to what environment?) that have long been recognized and debated in their respective disciplines. While an expanded version of our model may need to address these issues, they do not arise directly here.

<sup>2</sup> A similar assumption is present in Altenberg's (1994) model of constructional selection, where the number of genes increases over time, so it is possible for a genotype to be incomplete. We are not aware of any other NK-style models that allow for this possibility.

<sup>3</sup> As in the standard NK framework, the fact that our model allows a single variant per component (i.e., one allele per gene) is without loss of generality since the components could be defined in pairs or higher multiples (e.g., A, A', and

product architectures, as it enables products that deliver the same functions in different ways to coexist and compete with each other.

In our experiments, we construct a new component–function map and a new fitness landscape for every simulation run. Each component is assigned a pleiotropy  $k_i \in \{1 \dots F\}$  drawn independently from a discrete uniform distribution, then linked to  $k_i$  functions by sampling uniformly without replacement. Since the number of possible function values can be large (e.g., over a million for a function with a polygeny of 20), they are drawn “lazily” when needed and stored for the duration of the run.

### 3.2 Evolutionary Dynamics: Search and Selection

A simulation run consists of a fixed number of periods. In the first period,  $N$  firms are created with empty products. Each firm is activated in a random sequence. When activated, a firm engages in a local adaptive walk consisting of one or more steps. For each step, the firm chooses a component at random, and either adds the component to its product if the product does not already contain it or removes the component if it does. If the value of the product does not increase, the step is reversed and the walk ends. Otherwise the walk continues until a local peak is reached.<sup>4</sup>

Each subsequent period begins with population selection, which closely follows the procedure used by Levinthal (1997). Each firm is assigned a probability of survival equal to the ratio of its product’s value to that of the highest-valued product in the population. Firms that do not survive are replaced by new entrants. With probability equal to one minus the ratio of the average product value to the highest value in the population (a measure of population-level fitness called the genetic load), a new entrant starts with an empty product. Otherwise the entrant replicates the product design of a surviving firm, with the probability of a particular product being replicated equal to the ratio of its value to the total value in the population.

---

A” for three variants of A), with additional assumptions as appropriate (e.g., to ensure that all variants have the same pleiotropy set and/or that they are mutually exclusive in terms of the value they contribute to the product).

<sup>4</sup> We also experimented with “greedy” walks in which the firm evaluates more than one step and keeps the one with the greatest increase in value (if any), as well as walks that terminate after a maximum number of steps. Our results were generally robust to these variations in the local search procedure.

The intuition behind these assumptions is that as the population matures, its average value should increase relative to the maximum value (i.e., the genetic load should decrease), and entrants should find it more attractive to replicate an existing design than start the search process from scratch—whereas when overall fitness is low it should be more attractive to start from scratch and search in a different direction to avoid getting trapped on a local peak.<sup>5</sup>

The combination of local search and population selection yields a powerful set of evolutionary dynamics that enables firms to converge rapidly on high-value product designs. However, these dynamics can also lead to stagnation. If all firms' products have the same value (i.e., the ratio of every product's value to the highest value is one), they are all guaranteed to survive and new entry grinds to a halt. To maintain diversity in the population, we provide the ability for firms to engage in “long jump” search before their adaptive walk. A long jump entails swapping one or more components that are currently included in a firm's product for an equal number of components that are not, with all components chosen uniformly at random. (Each swap is equivalent to a two-bit mutation in a standard NK model.) As with local search, a long jump is only kept if it increases the value of the product, otherwise it is reversed and search process proceeds from the previous design. The number of long jumps attempted by a firm in a given period is drawn from a Poisson distribution with mean  $\lambda$ . The number of swaps in a given jump follows a geometric distribution, with a second swap having probability  $\mu$ , followed by a third (if the second occurs) with probability  $\mu$ , and so on until there are no more components to be swapped. For all of the results reported in section 4, we set  $\lambda = 1$  and  $\mu = 1/2$ .

### 3.3 Extensions

The assumptions stated above constitute the basic model. We studied three extensions to this model in order to explore the conditions that give rise to modular product architectures and promote the emergence of platform ecosystems. Formally specifying these extensions is

---

<sup>5</sup> Our population selection procedure differs from Levinthal's in that his random births yield completely random organizational forms, which would correspond to products containing a random assortment of components. We chose to have our non-replicating entrants start with empty products mainly because it seemed more plausible in our setting, and also because we wanted to avoid biasing the model in favor of product growth.



straightforward but cumbersome, so we describe them informally with an emphasis on their motivation and intuition rather than the precise modeling assumptions.

### 3.3.1 Nested Product Designs

Simon's (1962) famous parable of the watchmakers emphasized the importance of stable subassemblies in building modular systems. However, there is no obvious mechanism in the basic model to select and amplify such subassemblies, even if they were to arise by chance. To provide such a mechanism, we added the ability for firms to incorporate the products of other firms just as they would "primitive" components. When a firm adds a product as a subassembly, all of its components become part of the firm's product. The value of this product continues to be determined solely by the set of components it contains; duplicate components have no effect, nor do subassembly boundaries (i.e.,  $v(\{A, B\}) = v(\{\{A, \{A, B\}\}) = v(\{A, \{B\}\})$ ). But subassemblies retain their identity for purposes of local and long-jump search; they are added and removed as a unit. Moreover, if the firm changes the design of a product that has been incorporated into other products as a subassembly, then those products change accordingly—just as mid-tier suppliers in a complex supply chain may change the components of subsystems they supply to downstream firms (sometimes with adverse consequences, cf. Cabigiosu & Camuffo, 2012).

While our implementation of the model supports arbitrary levels of product nesting and complex (acyclic) supply chain topologies, we experimented primarily with a three-layered structure consisting of component suppliers, subsystem suppliers, and system integrators (also known as original equipment manufacturers, or OEMs). We denote the number of subsystem suppliers by  $S$ . Setting  $S = 0$  is equivalent to the basic model. When they are present, they only "see" primitive components for the purposes of their product design processes. OEMs, on the other hand, see both components and the products of the subsystem suppliers (but not the products of other OEMs). In other words, each layer has visibility into all of its upstream layers. Population selection occurs independently in each layer.

### 3.3.2 Environmental Change

Although modularity may confer advantages in a static environment, its benefits are magnified when the value of product may change over time due, for example due to changes in the way components interact with each other or the way functions are valued by consumers. To model environmental change, we added two parameters: the frequency of change ( $CF$ ), and the severity of change ( $CS$ ). The frequency of change is simply the probability that the environment changes in a given period, with zero corresponding to the basic model. If environmental change occurs, each function is “scrambled” with probability  $CS$ . If a function is selected to be scrambled, all of its function values are randomly redrawn (cf. Siggelkow & Rivkin 2005). 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 more narrowly targeted (like the substitution of fuel injection systems for carburetors).

### 3.3.3 Market Segmentation

Another well-known benefit of modularity is the ability to create economies of scope by making it easier to customize a product for different markets (Langlois & Robertson, 1992). We model the existence of multiple market segments by adding another two parameters to the model: the number of segments ( $MS$ ), and the correlation between them ( $MC$ ). The number of segments determines the number of distinct fitness landscapes that are created, with one corresponding to the basic model. Each firm is assigned a landscape at random. If  $MC = 0$ , each landscape is uncorrelated with the others; in other words, their fitness contributions are drawn independently and identically from the same distribution. If  $MS > 1$  and  $MC > 0$ , we define a “parent” landscape and  $MS$  “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, the same functions need to be inherited for each; we arbitrarily set the first  $F \cdot r$  functions to be inherited, and the remaining  $F(1 - r)$  to be unique.

## 4 Results

We implemented the model in Java using the MASON simulation toolkit (Luke et al., 2004), and conducted a preliminary set of experiments on a Linux-based high-performance computing cluster. Experiment 1 aimed to identify the factors that influence product modularity. Using the nested product design extension, we set the number of system integrators (OEMs) to 10 while varying the number of subsystem suppliers. We also varied the number of components and functions, obtaining results for all combinations of  $C = \{8, 16, 24, 32\}$ ,  $F = \{8, 16\}$ , and  $S = \{0, 5, 10\}$ . Experiment 2 focused on the environmental change extension, varying the frequency and severity of change ( $CF$  and  $CS$ ) in the range  $\{0, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1\}$ . In this experiment, we fixed  $C = 32$  and  $F = 8$  while varying  $S$  as in the first experiment. Experiment 3 was similar in structure to the second experiment but focused on the market segmentation extension. In this experiment, we varied the number of market segments ( $MS$ ) in the range  $\{1 \dots 5\}$  and the correlation of the segment landscapes ( $MC$ ) in the range  $\{1, \frac{3}{4}, \frac{1}{2}, \frac{1}{4}, 0\}$ , in addition to varying  $S$ . (Note that lower values of  $MC$  yield greater differences between landscapes.)

All of our reported results are averaged over 200 independent trials for each combination of parameters, with a different random seed for each trial. Each trial comprised 500 periods. The reported results are based on observations taken at period 200, but we ran the same analyses using observations at periods 100 and 500 to verify their robustness.

### 4.1 The Evolution of Modularity

Our first set of results addresses the central question of the paper: can modularity evolve? To answer this question, we use the measure of product modularity proposed by Frenken (2006). This measure is designed to have a minimum value of 0 for a product in which every component interacts with each other (i.e.,  $k_i = F$  for all  $i \in \{1 \dots C\}$ ), and a maximum

value of 1 for a product in which there are no interactions between components (i.e.,  $k_i = 1$ ).<sup>6</sup>

We need to modify the measure to account for the fact that in our setting, a product may contain only a subset of the available components and/or enable only a subset of the possible functions.

Letting  $|\mathbf{p}|$  denote the number of components in product  $\mathbf{p}$ , and  $k(\mathbf{p})$  denote the total number of functions enabled by the product, we define the modified measure of product modularity as follows:

$$M(\mathbf{p}) = 1 - \frac{1}{|\mathbf{p}|} \log_{k(\mathbf{p})} \prod_{i \in \mathbf{p}} k_i$$

(The measure is undefined for  $k(\mathbf{p}) < 2$ , since logarithms of base 0 and 1 are undefined. This is natural, as modularity is concerned with the way a product's functionality is partitioned across components; for fewer than two functions, there is no partitioning. Also note that a product with a single component and more than one function has a modularity of 0.)

Figure 2 illustrates the relationship between the number of available components, the size of a given product, and the modularity of that product. The contour plot clearly reveals that this relationship is positive: both higher values of  $C$  and larger products are associated with higher modularity. While the plot is shown for the basic model ( $S = 0$ ) and 8 functions, this pattern holds consistently across Experiment 1 (i.e., with  $S = \{5, 10\}$  and  $F = 16$ ).

Figure 3 focuses on the region of the parameter space with the highest modularity ( $C = 32$  and  $F = 8$ ), and provides a more detailed breakdown of its distribution by the number of subsystem suppliers. The box-and-whisker plot on the left side of the figure shows a median modularity of about 0.46 for  $S = 0$ . This is substantially higher than the expected modularity of a product composed of randomly selected components (about 0.36),<sup>7</sup> providing evidence that the evolutionary dynamics of our model favor modular product architectures.

That said, the mechanism by which we had hoped to increase the propensity for modularity to evolve—the nested product design extension—does not appear to have had the

---

<sup>6</sup> In an non-decomposable product (i.e., one in which no subset of components exclusively enables a subset of functions), the maximum modularity approaches 1 for a product in which every component has a pleiotropy of 1 except for a single component with pleiotropy  $F$  that serves as a technical standard or set of design rules.

<sup>7</sup> This result was obtained through numerical simulation, and depends on  $F$  but is independent of  $C$ . Additional numerical and analytical results on expected modularity will be provided in a subsequent version of the paper.

desired effect, since the median modularity is actually lower for  $S = 5$  and  $10$  than it is for trials in which there are no subsystem suppliers. A closer look at the data reveals why: almost every product in the subsystem layer enables all of the available functions (i.e.,  $k(\mathbf{p}) = 8$  for  $F = 8$  and  $k(\mathbf{p}) = 16$  for  $k(\mathbf{p}) = 16$ ). In other words, these products are not really subsystems at all but fully functional products in their own right. This is also reflected in the behavior of OEMs. In the vast majority of cases, OEMs simply adopt the product of the leading subsystem supplier—or assemble an identical product from raw components—rather than assembling a product from multiple subsystems.

We ran an additional set of experiments in which subsystem suppliers were specialized to a subset of the available components (e.g., for  $C = 32$  there might be 4 types of subsystem suppliers, each of which can only use 8 components). Here we also found strong incentives to enable as many functions as possible rather than make a product with limited functionality that complements other specialized subsystems. Of course, one way to induce this would be to limit the *functions* of the components available to the specialized subsystem suppliers, but this would be tantamount to assuming modularity, which is precisely what we want to avoid. Instead, we may need to revise our assumptions about the selection process to reward firms based on the value created by their products downstream. We experimented with these kinds of assumptions in earlier work, but left them out of this version to simplify the model (a reminder of Einstein’s dictum to make things as simple as possible but no simpler).

## 4.2 Modularity and Product Value

Turning to the relationship between modularity and product value, which we expect to be positive, we again find some supportive evidence but not the definitive result we expected. The supportive evidence comes from the linear regression analyses presented in Tables 1 and 2. They indicate that under both environmental change and market segmentation, modularity is positively related to product value. In these regressions, we control for all of the experimental parameters ( $CS$ ,  $CF$ ,  $MS$ ,  $MC$ , and  $S$ ) as well as three other firm-level attributes: period of entry,

size of the product (defined as the number of unique components), and number of nested products (0 if the product contains only raw components). This is encouraging news, as it provides a reason for more modular products to be selected and retained in the population, independent of their other characteristics.

Once again, however, further inspection reveals that the mechanism we expected to find is lacking. We anticipated a positive relationship between modularity and product value because it should be easier to find high-value products if they are modular than otherwise. By this logic, there should be a positive interaction between modularity and the model parameters that make search more difficult: frequency and severity of change in Experiment 2, and the number of market segments in Experiment 3 (as well as a negative interaction between modularity and the correlation between segments). In other words, as the difficulty of search increases, the marginal value of modularity should increase. But this is not borne out in our data. When we ran the regressions again with interaction terms, the coefficients were either negative or insignificant in almost all cases.

A possible source of explanation lies in the nature of the search process in our model. Unlike in the model of Frenken and Mendritzki (2012), which assumes parallel search within subsystems, our firms cannot engage in parallel search because we treat them as indivisible entities rather than organizations composed of individuals and teams. Moreover, we do not endow firms with the ability to identify subsystems, which is necessary in order to optimize them separately. The existence of subsystem suppliers *does* enable parallel search at the inter-firm level (since every firm engages in search in every period), but again it then becomes important to align the incentives of the firms in order to yield effective coordination.

### **4.3 Toward Platform Ecosystems**

While modularity is no doubt an important factor in catalyzing the emergence of a vibrant ecosystem of firms and products, we can observe this emergence more directly by examining the patterns of interdependence and reuse among components and products. We are

especially interested in identifying structures related to platform architectures, such as a densely connected core surrounded by a sparsely connected periphery (Murmann & Frenken, 2006; Baldwin & Woodard, 2009). “Core” components or products can become the basis for platform ecosystems by being frequently reused (i.e., appearing in many downstream products) and by supporting variety (i.e., appearing in combination with a wide range of other components or products).

To measure the overall tendency of an ecosystem to exhibit these characteristics, we constructed a measure called *average product centrality*. To compute this measure, 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 of actors, namely a weighted variant of the *group betweenness centrality* measure proposed by Everett and Borgatti (1999). 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 or subassemblies appear in products more often in combination with a particular product than with others.

Formally, consider a graph whose nodes are the full set of designs in a given simulation run (including raw components, subassemblies, and final products) and whose links indicate co-inclusion in a downstream design. For example, two components that appear in a subassembly would receive one link for that appearance, as would a component and a subassembly that appear in a final product. Let  $g_{u,v}(\mathbf{p})$  denote the number of geodesics (distinct shortest paths in the graph) that connect designs  $u$  and  $v$  while passing through designs in  $\mathbf{p}$ . Let  $g_{u,v}$  denote the total number of geodesics connecting  $u$  and  $v$ , and define an ordering ( $<$ ) on the set of components. Then:

$$\text{GBC}(\mathbf{p}) = \frac{1}{Q} \sum_{u < v} \frac{g_{u,v}(\mathbf{p})}{g_{u,v}} \text{ for } u, v \notin \mathbf{p}$$

where  $Q$  is a normalization factor equal to the maximum of the summation on the right, which depends on the size of the graph and the number of links.

Figures 4 and 5 summarize the average product centrality as a function of the key parameters in Experiments 2 and 3. Interestingly, centrality tends to increase along with the difficulty of search (although as the scale on Figure 5 indicates, it starts from a much lower base in Experiment 3). This finding suggests that more challenging environments lend themselves to platform-like architectures with stable cores and diverse peripheries. However, more analysis is needed to determine the mechanisms involved in these results. One puzzle is that we expected a lower level of centrality for extremely difficult environments, since we cannot see a clear benefit to having a platform architecture when it is effectively impossible to do much better than assembling a collection of components at random. In Experiment 2, we also expected an asymmetry between the effects of change frequency and change severity on average product centrality. Our intuition is that frequent but moderate changes should reward the ability to “mix and match” components and products to a greater extent than rare but catastrophic ones.

## **5 Conclusion**

In this paper, we sought to model the evolution of modular architectures and the emergence of platform ecosystems. While the model we presented does not quite succeed at doing either of these things, it does yield some promising findings which we hope can serve as a baseline for future work. The main problem to be solved is that while it may be advantageous for our firms to design modular products, they do not do so nearly as often or as convincingly as we would like. We suspect that this can be remedied by a combination of deeper analysis (e.g., making sure we understand the structure of the fitness landscapes adequately), changes to the model (e.g., selecting subsystem suppliers based on the value their products create downstream rather than comparing them to their peers in the same layer), and additional experimentation (e.g., exploring larger design spaces, collecting full time series data instead of just snapshots). We look forward to undertaking these tasks in the coming months.

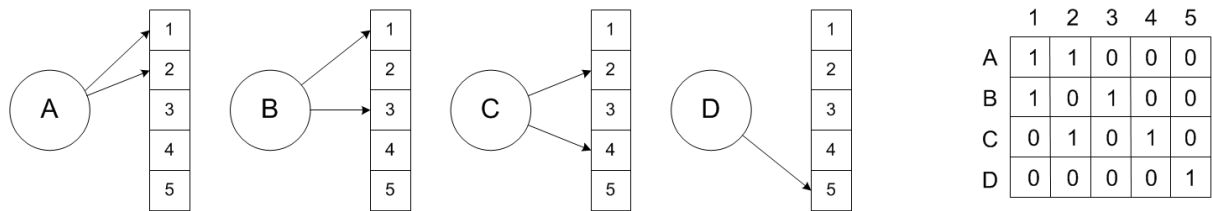


## References

- Altenberg, L. 1994. Evolving better representations through selective genome growth. In *Proceedings of the IEEE World Congress on Computational Intelligence*, Pt. 1, 182–187.
- Arthur, W. B. 2009. *The Nature of Technology: What It Is and How It Evolves*. Free Press.
- Baldwin, C. Y., & Clark, K. B. 1997. Managing in an age of modularity. *Harvard Business Review*, 75(5): 84–93.
- Baldwin, C. Y., & Clark, K. B. 2000. *Design Rules: The Power of Modularity*, Vol. 1. MIT Press.
- Baldwin, C. Y., & Woodard, C. J. 2009. The architecture of platforms: A unified view. In A. Gawer (Ed.), *Platforms, Markets and Innovation*: 19–44. Edward Elgar.
- Beinhocker, E. D. 2006. *The Origin of Wealth: Evolution, Complexity, and the Radical Remaking of Economics*. Harvard Business School Press.
- Brusoni, S., & Prencipe, A. 2001. Unpacking the black box of modularity: Technologies, products, and organizations. *Industrial and Corporate Change*, 10(1): 179–205.
- Brusoni, S., Marengo, L., Prencipe, A., & Valente, M. 2007. The value and costs of modularity: A problem-solving perspective. *European Management Review*, 4(2): 121–132.
- Cabigiosu, A., & Camuffo, A. 2012. Beyond the “mirroring” hypothesis: Product modularity and interorganizational relations in the air conditioning industry. *Organization Science*, 23(3): 686–703.
- Dosi, G., & Marengo, L. 2005. Division of labor, organizational coordination and market mechanisms in collective problem-solving. *Journal of Economic Behavior and Organization*, 58(2): 303–326.
- Ethiraj, S., & Levinthal, D. A. 2004. Modularity and innovation in complex systems. *Management Science*, 50(2): 159–173.
- Evans, D. S., Hagi, A., & Schmalensee, R. 2006. *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*. MIT Press.
- Everett, M. G., & Borgatti, S. P. 1999. The centrality of groups and classes. *Journal of Mathematical Sociology*, 23(3): 181–201.
- Frenken, K. 2006. A fitness landscape approach to technological complexity, modularity, and vertical disintegration. *Structural Change and Economic Dynamics*, 17(3): 288–305.
- Frenken, K., & Mendritzki, S. 2012. Optimal modularity: a demonstration of the evolutionary advantage of modular architectures. *Journal of Evolutionary Economics*, 22(5): 935–956.
- Gawer, A., & Cusumano, M. A. 2002. *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*. Harvard Business School Press.

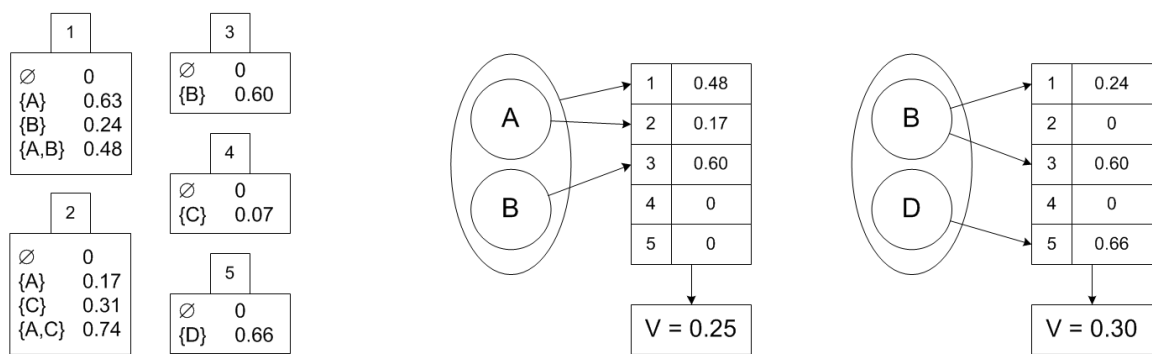
- Iansiti, M., & Levien, R. 2004. *The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability*. Harvard Business School Press.
- Jacobides, M. G. 2005. Industry change through vertical disintegration: How and why markets emerged in mortgage banking. *Academy of Management Journal*, 48(3): 465–498.
- Kauffman, S. A., & Levin, S. 1987. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128(1): 11–45.
- Kauffman, S. A., & Weinberger, E. D. 1989. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology*, 141(2): 211–245.
- Kauffman, S. A. 1993. *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press.
- Langlois, R. N., & Robertson, P. L. 1992. Networks and innovation in a modular system: Lessons from the microcomputer and stereo component industries. *Research Policy*, 21(4): 297–313.
- Levinthal, D. A. 1997. Adaptation on rugged landscapes. *Management Science*, 43(7): 934–950.
- Luke, S., Cioffi-Revilla, C., Panait, L., & Sullivan, K. 2004. MASON: A new multi-agent simulation toolkit. In *Proceedings of the Eighth Annual Swarm Users/Researchers Conference (SwarmFest 2004)*.
- Marengo, L., Dosi, G., Legrenzi, P., & Pasquali, C., 2000. The structure of problem-solving knowledge and the structure of organizations. *Industrial and Corporate Change*, 9(4): 757–788.
- Murmann, J. P., & Frenken, K. 2006. Toward a systematic framework for research on dominant designs, technological innovations, and industrial change. *Research Policy*, 35(7): 925–952.
- Rivkin, J. W., & Siggelkow, N. 2007. Patterned interactions in complex systems: Implications for exploration. *Management Science*, 53(7): 1068–1085.
- Siggelkow, N., & Rivkin, J. W. 2005. Speed and search: Designing organizations for turbulence and complexity. *Organization Science*, 16(2):101–122.
- Simon, H. A. 1962. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6): 467–482.
- Ulrich, K. 1995. The role of product architecture in the manufacturing firm. *Research Policy*, 24(3): 419–440.

# Figures and Tables



## (1) Component-function map

Which components enable which functions?



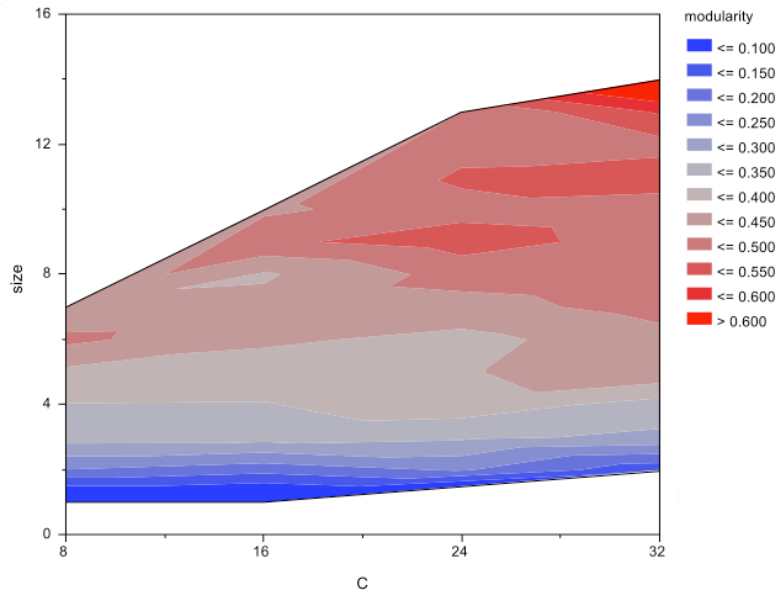
## (2) Function value

How well do different component combinations enable each function?

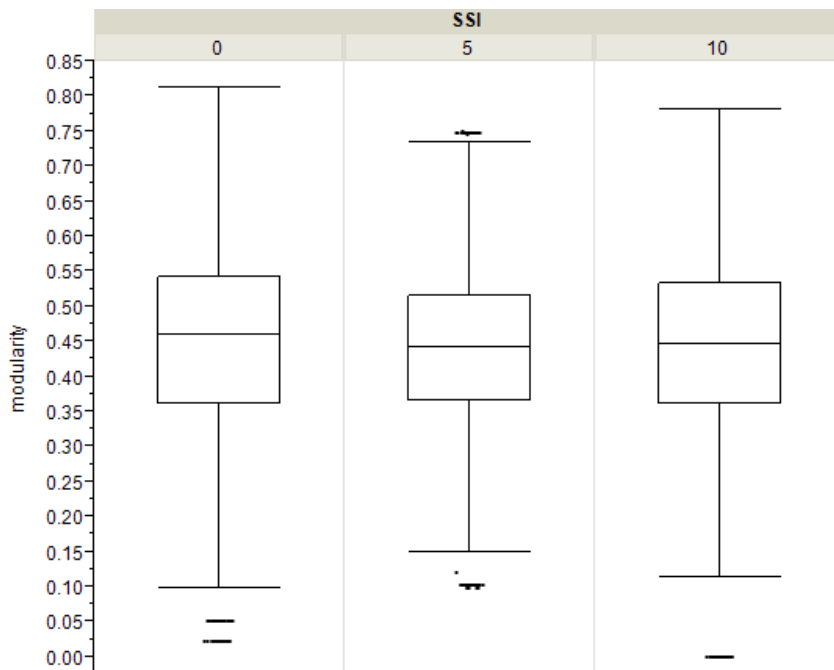
## (3) Product value

What is the overall value (fitness) of a given set of components?

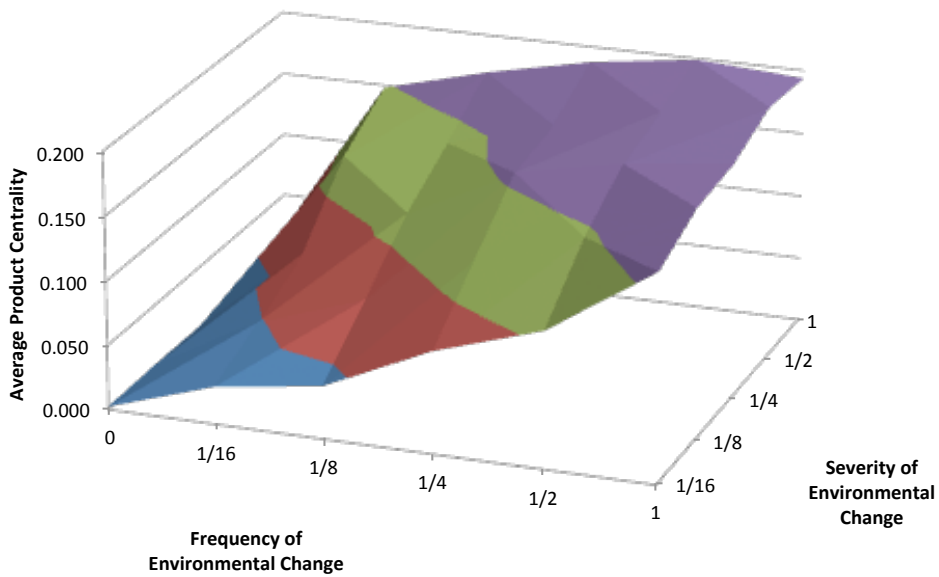
**Figure 1. Constructing a rugged fitness landscape for product designs**



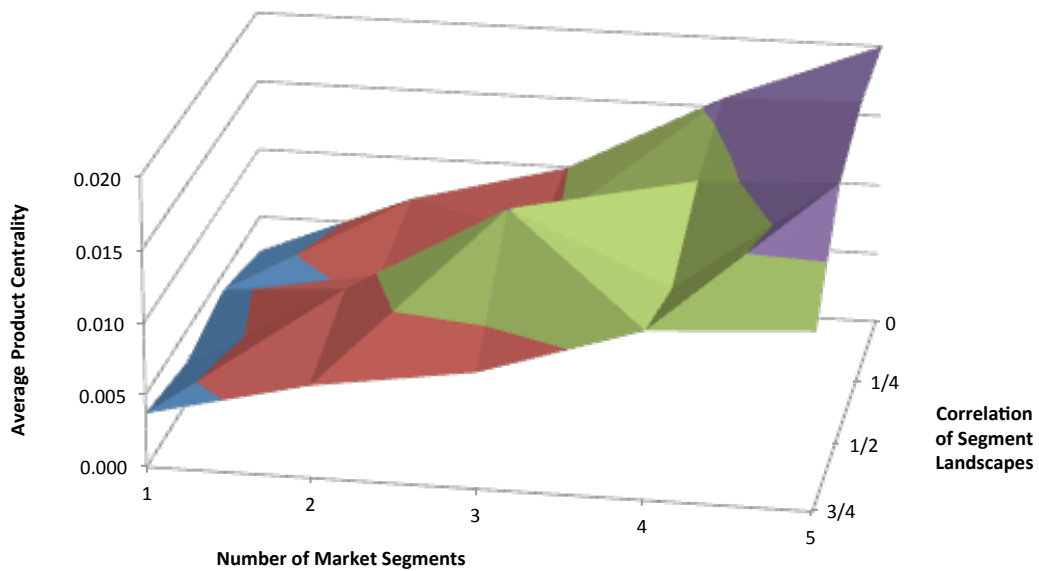
**Figure 2. Contour map of product modularity by number of components and product size (Experiment 1,  $SSI = 0$ ,  $F = 8$ )**



**Figure 3. Distribution of product modularity by number of subsystem suppliers (Experiment 1,  $C = 32$ ,  $F = 8$ )**



**Figure 4. Average product centrality by frequency and severity of environmental change (Experiment 2,  $SSI = 5$ )**



**Figure 5. Average product centrality by number of market segments and correlation of segment landscapes (Experiment 3,  $SSI = 5$ )**

**Table 1. OLS regression results for environmental change (Experiment 2)**

. regress value entered size nested modularity s cs cf if c==32 & f==8

Source	SS	df	MS	Number of obs =
Model	1301.47088	7	185.924411	179927
Residual	1711.9707179919		.00951523	F( 7,179919) =19539.67
Total	3013.44157179926		.016748227	Prob > F = 0.0000
				R-squared = 0.4319
				Adj R-squared = 0.4319
				Root MSE = .09755

value	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
entered	-.0002849	1.81e-06	-157.57	0.000	-.0002884 -.0002814
size	.0028289	.0000654	43.28	0.000	.0027008 .002957
nested	-.0164763	.0005536	-29.76	0.000	-.0175613 -.0153913
<b>modularity</b>	<b>.0970528</b>	<b>.0020086</b>	<b>48.32</b>	<b>0.000</b>	<b>.0931159 .1009896</b>
s	.0003808	.0000652	5.84	0.000	.000253 .0005085
cs	-.0914885	.0006831	-133.94	0.000	-.0928273 -.0901497
cf	-.1359076	.0007499	-181.24	0.000	-.1373773 -.1344378
_cons	.8497189	.0012486	680.54	0.000	.8472717 .8521661

**Table 2. OLS regression results for market segmentation (Experiment 3)**

. regress value entered size nested modularity s ms mc if c==32 & f==8

Source	SS	df	MS	Number of obs =
Model	17.0473812	7	2.43534017	119998
Residual	152.880326119990		.001274109	F( 7,119990) = 1911.41
Total	169.927708119997		.0014161	Prob > F = 0.0000
				R-squared = 0.1003
				Adj R-squared = 0.1003
				Root MSE = .03569

value	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
entered	-.0000514	8.18e-07	-62.85	0.000	-.000053 -.0000498
size	.0022688	.0000459	49.41	0.000	.0021788 .0023588
nested	.0075612	.0002762	27.37	0.000	.0070198 .0081026
<b>modularity</b>	<b>.0367332</b>	<b>.0008695</b>	<b>42.25</b>	<b>0.000</b>	<b>.0350291 .0384374</b>
s	-.000131	.0000283	-4.63	0.000	-.0001864 -.0000755
ms	.0048127	.0000748	64.31	0.000	.004666 .0049594
mc	.0102676	.0003721	27.60	0.000	.0095384 .0109968
_cons	.817252	.0005328	1534.02	0.000	.8162078 .8182962