

Functional lock-in and the problem of design transformation

Andy Dong¹

Received: 24 September 2015/Revised: 19 July 2016/Accepted: 25 July 2016/Published online: 9 August 2016
© Springer-Verlag London 2016

Abstract The act of introducing an innovation into an existing product by substituting or inserting new technologies is thought to be challenging due to the problem of integrating new components and sub-system architectures into existing ones. This article aims to challenge the foundation of this problem and develop new insights into the choice of functional architecture. The article will propose that the choice of functional architecture to achieve an intended purpose locks-in a design by influencing the cost of transformation. This paper studies functional lock-in based on the transformation cost of the functional architectures of products. The transformation cost for a set of biological and biologically inspired products is compared to that of engineered products. The results show that the biological and biologically inspired products have a statistically significant lower transformation cost than the engineered products. The results indicate that the structure of functions and flows in a product will constrain its transformation. More broadly, the paper proposes minimum transformation cost as an essential property of an optimal design.

Keywords Product architecture · Innovation · Dominant designs · Graph edit distance

1 Introduction

A practical challenge associated with engineering design innovation is transforming an existing design (a product's components and their configuration) into a new design that employs new solution principles. The product is not conceptually new; it retains its intended purpose, but the product employs some new technologies and associated componentry configured in a new (but not wholly new) way. The Apple iPhone introduced in 2007 marked the start of the smartphone with a touch screen interface as a dominant design for cell phones, but the product was nonetheless a cell phone.

The changes may come about due to any number of causes, including new requirements driven by customer demand and revised design specifications (Fernandes et al. 2015) or corrections of design errors (Sudin and Ahmed-Kristensen 2011). Rather than seeking a new way to achieve a given function, engineering designers introduce incremental improvements to a product over time. These improvements may *eventually* result in a wholly transformed product employing new solution principles and technologies, but incremental improvements avoid as many changes as possible (Eckert et al. 2012; Jarratt et al. 2011).

This is a challenge encountered most prominently by companies with 'legacy' designs. Examples of these companies include the major automotive companies (e.g., Ford, Fiat, Toyota) as they try to develop electric vehicles, compared to Tesla, or software companies such as Microsoft as they struggle to transform their legacy desktop computing products into products for mobile computing, compared to Google or any number of startups capitalizing on the mobile computing trend. The incumbents will make gradual changes to a product, such as Microsoft adding new collaboration features to the Office Suite to mimic

✉ Andy Dong
andy.dong@sydney.edu.au

¹ Faculty of Engineering and Information Technologies, The University of Sydney, Sydney, NSW 2006, Australia

Google's 'Apps for Work' platform, but the product and functional architecture will largely remain the same.

The problem described above is named 'design innovation by transformation.' Design innovation by transformation is the process of changing a product's solution principles (Hubka 1992) and associated components to achieve the same primary function of the product at higher levels of performance. This wording borrows from the scholarship of innovations in design through transformations (Singh et al. 2009). In that conceptualization, design innovation *through* transformation is defined as a product that can change its physical configuration (or state) in operation to achieve a new primary function. In the current sense, transformation is defined as a change to the existing components and their architectural configuration at design time to adopt one or more new solution principles.

The problem of design innovation by transformation overlaps with the research scholarship of two areas. The first area addresses the problem of designing a product to accommodate future changes, synonymously called design for flexibility (Neufville and Scholtes 2011) or design for changeability (Niese and Singer 2014).

The second area addresses the technology infusion problem (Denman et al. 2011; Suh et al. 2010; Smaling and Weck 2007). In technology infusion, the problem is determining the cost of implementing alternative architectures into an existing product architecture, including 'downstream' organizational change costs. Similarly, research in estimating the cost of change propagation takes into consideration the degree and risk of architectural change that a new innovation may introduce into an existing system architecture (Giffin et al. 2009; Clarkson et al. 2004). Underlying this body of influential scholarship is the conceptualization of products (or systems) based upon their physical architecture (Ulrich 1995). While it is not debated that the physical architecture imposes constraints upon the flexibility of the product to change, such a conceptualization nonetheless suffers from several shortcomings. First, scholars cannot agree at what level of architecture the analysis of flexibility should take place. Should the analysis take place at the level of systems and sub-systems, which are modules of systems and components (Mikkola and Gassmann 2003), respectively, or at the level of components and their interfaces? In fact, there is no literature providing a theoretical basis for the 'right' level of granularity for architectural decomposition (Chiriac et al. 2011). There is no agreed-upon metric of architecture modularity for engineered products (Sarkar et al. 2014; Hölttä-Otto et al. 2012). The lack of consistency leads to ambiguous strategic advice in directing design search to improve the flexibility of the product.

Second, this architecture paradigm is based upon the theory that the choice of technology and its components

create lock-in through component interdependencies. Small shifts in one part of the product cannot be made without making accompanying changes to other parts of the design. New technologies may be difficult to implement into an existing architecture due to component or interface incompatibility. Such a theory obscures knowledge as an impediment to the integration of a component into an existing architecture. Architectural integration is not only a simple technical matter of connecting physical interfaces. The Boeing 787 introduced a number of technological innovations into civilian aircraft such as the large-scale use of composite materials rather than aluminum alloys (Toensmeier 2005). While the technologies did not create fundamental changes to the architecture of the plane and were believed to be isolatable to their respective sub-systems, their introduction introduced new knowledge interdependencies that could not have been known beforehand (Kotha and Srikanth 2013). Physical architecture-based models of design innovation tend to underplay the influence of the structure of knowledge of the product. The structure of the knowledge may impose a hidden cost that is not directly observable from the physical architecture alone.

This research is intended to provide a new perspective that addresses the above shortcomings. First, rather than focusing on the physical architecture, the paper argues for a focus on the underlying knowledge structure. The paper will rely on the concept of functional models as a representation of a product's knowledge structure. The paper presents the case that constraints on flexibility occur before the establishment of a physical architecture. They appear once engineers commit to a set of solution principles, which will be reflected in the functional architecture for the product. To illustrate these ideas, the paper conducts a comparative, empirical analysis of a set of engineered products and biological systems. To perform the comparison, the paper lays down the foundation for a theoretically consistent empirical methodology to analyze the constraints on transformation as a graph edit distance (GED) problem (Gao et al. 2010). The GED provides the mechanism to compute the cost of transformation from one graph structure to another, from the existing functional architecture to another. The paper proposes a uniform basis for the comparison of the cost of transformation through comparison with two important graph structures—the small-world network structure (optimal for diffusion of information due to high clustering and short average path lengths) and tree-like structures (minimal structure to connect a set of nodes). This research will depart from approaches that seek to understand the degree of change available to a product based upon the propagation of changes through its physical, architectural configuration. Rather, the paper explores this question based upon the

structural transformation of a product's functional architecture.

2 Background and theoretical development

2.1 Design for innovation

Scholarship in understanding the flexibility or changeability of existing products (Hu and Cardin 2015; Cardin et al. 2013; Suh et al. 2007) is motivated by the reality that 'new' engineering design in industry is rarely a novel *de novo* design; rather, a new product more commonly arises through changes to existing products (Jarratt et al. 2011). This article will focus on changes to the functional architecture of the product, while acknowledging that a number of other associated changes such as process (how the product will be made) and management (the organization of the design team) coincide in the overall analysis of the changeability of a product (Niese and Singer 2014).

One common design strategy to improve flexibility is to standardize interfaces between elements and group elements into modules or sub-systems. Products that have a modular architecture would be amenable to rapid changes because a change in any one module would not necessarily require a cascade of changes to the product due to standardized interfaces between modules. A highly modular product architecture has been shown to decrease the time to design the product (Kong et al. 2009), support end-user innovation (Meehan et al. 2007), and facilitate the establishment of product platforms and families (Cai et al. 2009) among other benefits that increase the flexibility to change (Gershenson et al. 2003).

When the engineering change includes a new technology that is inserted into an existing product architecture (Denman et al. 2011), the transformation entails changes to the physical architecture and ensuing changes to the flow of material, energy, and information between components (Suh et al. 2010). These seemingly minor and incremental improvements are nonetheless an important source of continual product innovation (Banbury and Mitchell 1995); even a momentary gap in continuous improvement can have a significant effect on long-term competitiveness (Talay et al. 2014).

As it is now generally acknowledged that a product's physical architecture imposes costs on change, in what could be termed a 'design' turn in forecasting, scholars have been creating models to forecast the potential for design innovation given an existing design. These models aim to predict the long-term future for a product given its current structural components and their interconnections (Fujimoto 2014). These models are based on the design of products rather than on cumulative production (e.g.,

learning curve) or consensus expert opinion. At least two architectural properties matter in the analysis of the potential for design innovation: (a) the number of elements in the physical architecture of the design; and (b) the coupling of the elements. A high degree of coupling between design targets and design variables (Suh 2001) or between properties of the designer–artifact–user system (Maier and Fadel 2006) increases the challenges associated with making changes to either the design targets or the design variables because changes to the former will invariably alter the latter. At least one process-oriented property matters, the number of steps associated with the process that transforms a design problem into a designed product (Summers and Shah 2010).

One method to forecast this potential takes into account the maximum design complexity d^* . The design complexity is a metric for the number of components with which the slowest-improving component has dependencies (McNerney et al. 2011). According to this model, reducing the design complexity of the product architecture should increase its rate of improvement. Designs are complex and considered costly to change if their architectures have highly integral connectivity (McNerney et al. 2011). In other similar models, pairwise dependencies capture the likelihood and impact of risk of change (Eckert et al. 2011; Giffin et al. 2009). Architectural complexity is more likely to be related to a strategic decision about the degree of centralization or distribution of the product architecture (Sinha and de Weck 2013) rather than an intrinsic feature of the product. Furthermore, one of the challenges in enacting the recommendation to reduce design complexity by reducing the number of dependencies between components is that the connectivity of an engineered system is sometimes determined by the 'natural' existence of hubs in its architecture (e.g., Sosa et al. 2011). For example, wire bundles tend to connect to a single power distribution panel, which cannot be practically eliminated. It is also possible for system architectures to have a highly non-homogeneous connectivity, that is, a few components connect to many other components, but most components are sparsely connected, resulting in a system architecture of low complexity. Bus-like architectures, such as those encountered in computer hardware, have high connectivity between the bus to 'leaf' nodes, which are generally modular sub-systems that are loosely coupled to the rest of the system, resulting in an architecture of low complexity (Höltkä-Otto et al. 2012). In other words, the connectivity of components is the result of both strategic design decisions and the intrinsic technical characteristics of some products, which may have no relation to the core performance of the product that companies seek to improve through incremental improvement. The architecture of a product can be changed to reduce its complexity. The

decision to do so is as much a consequence of business strategy as it is a consequence of engineering design decisions. Therefore, the complexity of the product architecture may not provide accurate guidance about the potential for design innovation, especially if the innovation would involve changes to the solution principles. In particular, the architecture perspective ignores the cost of knowledge change as companies reallocate assets and resources (Sanchez and Mahoney 1996).

Henderson and Clark (1990) studied the often under considered cost of knowledge change when companies introduce innovations into existing product architectures, which they described as architectural innovation. They examined the relationships between architectural knowledge as embodied in the product architecture and the capability of companies to implement architectural innovation. They showed that companies are often challenged to change their information processing behaviors to suit changes in product architecture—exactly the type of architectural changes studied previously (Denman et al. 2011; Suh et al. 2010). They found, for example, that a modular physical product architecture does not necessarily lead to a simplified, modular knowledge structure wherein each designer knows exactly how their knowledge interfaces and interacts with others' knowledge. Brusoni and Prencipe (2001, pp. 201–202) emphasize the point 'that product modularization does not derive from, nor bring about, knowledge modularization.' A significant amount of knowledge is embodied in the product and in the organizational structure. An existing product and its knowledge structure embody an organization's cumulated knowledge. Even a subtle change to the architecture can pose a threat to the organization with the incumbent design. In essence, their contention is that there exists a knowledge structure underlying a product. Architectural changes introduce transformation costs by forcing the realignment of organizational boundaries to match the physical architecture of the product (Sanchez and Mahoney 1996). That long-established engineering companies do not change the basic underlying function structure of their product even as they introduce new component or architectural innovations to achieve the same overall functionality suggests that functional lock-in happened long before a stable architecture emerges (Wyatt et al. 2009).

A product's knowledge structure must therefore necessarily influence the dynamics of change for the product. The knowledge structure encodes the path dependency, that is, a commitment to the solution principles of a particular product (Dosi 1982). In other words, suppose that there are two available technologies to accomplish a function for a product, but the rate of progress of the technologies is unknown. Examples include lithium-ion batteries or hydrogen as alternative energy sources for cars, or solar

energy from crystalline- or amorphous-silicon technology. While the products may have similar physical architectures, their underlying knowledge structures can differ significantly due the choice of technology. This difference may impact design innovation by transformation. Thus far, the knowledge structure has not been considered as a factor influencing the cost of design transformation. Where knowledge has been considered as part of calculating the cost of change, the knowledge exists in the form of prior cases to solve functional and component conflicts (Fei et al. 2011). Therefore, we know precious little about the constraints on change that the *knowledge structure* of a product can impose. There may be functional architecture structures that are more or less optimal in relation to the degree to which the structure imposes costs on design innovation by transformation.

2.2 Architectural properties of optimal designs

This article is not the first to consider the structural properties of the functional architecture of a product and its relation to the optimality of a product's design. Suh (2001) is one of the first scholars to theorize principles of optimal designs from a knowledge perspective. In axiomatic design, the independence axiom states that the best design is the one that partitions functional requirements (FRs) and design parameters in such a way that a design parameter can be modified to suit its associated FR without affecting any other FR (Suh 2001, p. 16). That is, the FRs of the product should be independent of each other. In a mathematical model, the FR and the parameter (DP) represented in a matrix \mathbf{A} , which Suh calls the design matrix as shown in Eq. 1, should be linearly independent.

$$\begin{pmatrix} \text{FR}_1 \\ \text{FR}_2 \\ \vdots \\ \text{FR}_n \end{pmatrix} = \begin{matrix} & \mathbf{A} & \\ & \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix} & \begin{pmatrix} \text{DP}_1 \\ \text{DP}_2 \\ \vdots \\ \text{DP}_n \end{pmatrix} \end{matrix} \quad (1)$$

When the FRs of the product are independent, each FR is satisfied by exactly one design parameter, and there is an exact correspondence between each FR and design parameter. Any function will not affect another function; a design parameter satisfying a function can be altered and improved without affecting another function. All the a_{ij} except for 1 in each row will be 0, and only 1 a_{ij} in each row will be either 1 or some positive dependence value. Then, \mathbf{A} , properly permuted or reordered, would be a diagonal matrix. The independence axiom can also be expressed in a directed

bipartite graph form. Let $\mathcal{A} = (V, E)$ where $V = V_1 \cup V_2$, and each edge has the form $e = (x, y)$ where $x \in V_1$ and $y \in V_2$. Each type-1 node x in \mathcal{A} is referred to as an FR, and each type-2 node y is referred to as a design parameter. Then, the independence axiom requires that, e.g., $FR_1 \rightarrow DP_1$, $FR_2 \rightarrow DP_2$. Each node has a degree¹ of exactly 1.

This principle can be extended to functional models. Functional models link the different functional modeling perspectives relevant to the different disciplines associated with the design of engineered products (Eisenbart et al. 2013). The type of functional model that will be used in this paper is the functional basis (Hirtz et al. 2002; Stone and Wood 2000), an approach to functional modeling based upon the Pahl and Beitz (1999) function structure approach. A functional model can be represented as a directed bipartite graph $\mathcal{G} = (V, E)$ where $V = V_1 \cup V_2$, and each edge has the form $e = (a, b)$ where $a \in V_1$ and $b \in V_2$. Each type-1 node a in \mathcal{G} is referred to as a function, and each type-2 node b is referred to as a flow. According to the principle of independent functions, the ideal functional model should have independent functions, that is, $function_1 \rightarrow flow_1$, $function_2 \rightarrow flow_2$, exactly as with the graph representation of the design matrix. In practice, no product can attain this ideal functional model. Functional models generally have at least two functions connected by a common flow, e.g., $function_1 \rightarrow flow_1$, $function_2 \rightarrow flow_1$. Nonetheless, the principle is that an ideal functional model could attain a node degree of exactly 1. Research in the technological progress potential of energy harvesting products (Dong and Sarkar 2015) shows exactly this finding: as the node degree in a functional model decreases toward 1, and similarly the clustering coefficient tends toward 0, the progress potential for the respective product increases because of decreasing functional dependencies. However, a node degree of exactly 1 for every node in \mathcal{G} is physically meaningless in terms of products, and accordingly, their knowledge structures, because this would result in $n/2$ disconnected sub-graphs, where n is the total number of type-1 nodes (functions). There has to be a sufficient number of edges (flows) connecting the functions into a single, connected graph with no disconnected nodes. The most parsimonious functional model would have the fewest number of nodes and edges connecting the functions and flows into an aggregation to perform a certain intended purpose. That is, there should be a physically meaningful answer to the question, ‘Given a set of functions, how few edges can there be between them?’

There are two structures that would be the most parsimonious. The first structure is a tree. For a given set of nodes n , the structure that would have the minimum

number of edges forming a connected graph is a tree, which has exactly $n - 1$ edges. In software architecture, the tree is one of the most important data structures due to its efficiency for a range of computing processes, such as an index lookup. While the tree structure is minimal in terms of number of edges, the paths between the nodes can become long. This introduces inefficiencies in the transmission of energy, material, or information between function nodes. However, if there is only a requirement to move energy, material, or information up or down one level in a hierarchy, then this is the most efficient structure.

A second structure is the small-world network. In all networks, as the number of edges decreases, the more sparse the network becomes. It is known, though, that even in very sparse networks, the small-world properties of low average path distance allow for efficient global information transfer (Watts 1999). In small-world networks, connections between nodes are highly clustered. That is, any node is likely to be connected to another node in its cluster or neighborhood, no matter how this is defined, while, at the same time, the average number of intermediate nodes needed to connect any two nodes across the network remains relatively short. The unusual combination of high clustering and short path lengths in the same network turns out to be an important organizing principle for increasing the performance in many different types of systems. Collaboration networks, a type of social network, tend to organize into small-world networks because the small-world network properties enable the efficient exchange of information between collaborators (Uzzi and Spiro 2005). Likewise, studies of large-scale product development task networks demonstrate that effective task networks are organized into small-world networks because the small-world topology enables rapid exchange of information between tasks (Braha and Bar-Yam 2007). The small-world network may be another possible parsimonious structure for the functional model of a product. The addition of each extra edge to the structure may provide more flexibility in energy, material, or information exchange but also has associated costs of structure formation and maintenance. In a small-world network, there are ‘just enough’ edges in the knowledge structure to ensure efficient exchange of energy, material, and information. Therefore, the condition on small-world networks can provide a lower bound on the theoretical question of how few edges there can be in a functional model given a set of functions.

A small-world network is characterized by several metrics. The clustering coefficient in a small-world network is much higher than in an equivalent random network. The clustering coefficient is defined in two ways. The standard ‘social network’ definition of the (global) clustering coefficient for a network (Newman 2010) is given by Eq. 2:

¹ The node degree is the number of edges connected to the node.

$$C = \frac{\text{number of triangles} \times 3}{\text{number of connected triples}} \quad (2)$$

Watts and Strogatz propose an alternative local clustering coefficient measure (Watts 1999). The Watts–Strogatz local clustering coefficient for node i is given by:

$$C_i = \frac{2E_i}{k_i(k_i - 1)} \quad (3)$$

where E_i is the number of edges that actually exist and k_i is the number of edges connecting node i – k_i other nodes. The second metric is the path length, or the number of ‘hops’ between nodes, which is expected to be much smaller than the diameter of the network, the maximum number of ‘hops’ across the entire network. The path length reported is the characteristic path length ρ between nodes in a small-world network (Watts 1999). To assess the small-worldness of the functional models, this paper uses the method proposed by Humphries et al. (2006), which relies on comparing the ρ and C values of a network with a random network that has a similar number of vertices and average node degree. The small-world index (SWI) of each network can then be calculated using the following equation:

$$\text{SWI} = \frac{C\rho^R}{C^R\rho} \quad (4)$$

It is sufficient to show small-worldness if the global clustering coefficient is larger than the clustering coefficient of an equivalent random network with the same number of nodes and mean node degree (Humphries et al. 2006), or equivalently, its SWI, the ratio of a network’s clustering coefficient to that of an equivalent random network, is >1 . Even if the knowledge structure of a product satisfies the small-world network properties, there would still be a cost incurred to transform it to another small-world structure. If, however, both the global and Watts–Strogatz clustering coefficient are 0, then it implies that the functional model is tending toward a tree-like structure. Tree-like structures have zero clustering coefficients. If the graph structure approaches a tree-like structure, then it will have close to zero clustering coefficient.

To sum up, this section discussed and applied Suh’s independence axiom to consider the optimal structure of functional architectures. While no feasible functional architecture could satisfy the independence axiom, this section explained why tree-like and small-world graphs should be the most parsimonious structure for feasible functional architectures. These two important structures can therefore be used as idealized functional architectures to which real-world engineered products and biological systems could be transformed. The transformation cost from a functional architecture with a given structure to

these two structures provides a uniform basis for comparison across products.

2.3 Biological systems

An empirical challenge in performing this research lies in identifying a set of products for analysis and comparison. What ‘products’ might set the benchmark for lowest transformation cost? For examples of these products, this paper uses biological systems² because biological systems are assumed to be optimal or at least nearly optimal.³ Indeed, this is one of the attractions of biomimetic design: Nature provides examples of (nearly) optimal designs that can be copied to achieve an intended purpose. In this context, nature’s products embody two types of optimality. First, the biological system contributes to a species’ optimal performance within its particular environment and way of living relative to competing species. Weaker competing species eventually die out because natural selection tends to eliminate poor designs. In addition to optimal performance, biological systems are likely to have an optimal functional architecture. Their functional architecture might be optimal in two senses. First, the functional architecture would have the most parsimonious number of functions and flows of material, energy, and information to achieve an intended purpose. In a study comparing the way that technology and nature address a similar problem (Vincent et al. 2006), it was found that in engineering the path to a solution tends to favor changing the type or amount of material and accordingly the energy requirement. In terms of functional architecture, this solution path results in increasing the number of types of energy nodes and the number of types of material edges. In contrast, in nature, there is an economy of design. Intended purpose is achieved by manipulations of shape and combinations of materials at larger sizes, with larger sizes themselves achieved by high levels of hierarchy, while economizing on the number of different functions. Second, it would be advantageous for biological systems to have functional architectures that are the least costly to transform into another functional architecture. The new functional architecture is necessary to achieve new intended purposes or levels of performance that are required for the species to adapt to the changing environment. Since nature (re)designs through evolution, it is plausible to assume that the functional architecture of nature’s designs should impose the least cost on potential transformations. A functional architecture that imposes few impediments on transformation

² This paper uses the term *biological systems* to refer to plant or animal species or parts thereof, but not ecosystems.

³ It is debated that some of nature’s designs are not optimal, such as the human visual cortex. This article assumes, though, that in general nature’s designs, biological systems, are optimal.

(has the least functional lock-in) could be transformed into another functional architecture through the least number of steps (generations) compared to other counterparts. An optimal functional architecture would therefore require the least number of generations for a species to change from one form to another. This assumption is consistent with the documentation of substantial micro-evolutionary changes on the timescale of a few generations by evolutionary biologists (Gingerich 2001; Uyeda 2011).

In sum, biological systems should have optimal designs for achieving an intended purpose. First, they have the best level of performance relative to competing species for survival in their environment. Second, they have parsimonious functional architectures. Their functional architectures have: (1) the minimum number of ‘functions’ necessary to achieve an intended purpose; (2) the minimum number of flows of material, energy, and information (signals); and (3) their functional architectures are likely to impose the least cost on transformations into another functional architecture.

3 Hypothesis

The proposition tested by the hypotheses is that when a product undergoes design innovation by transformation, its knowledge structure is transformed. This article claims that the existing knowledge structure of a product locks-in a design and constrains the selection of components and component architecture. The existing knowledge structure for the product may impose constraints on the flexibility of change since components and component architecture must satisfy the underlying function and flow requirements. The product’s knowledge structure, represented by a functional model, therefore has structural characteristics that affect the flexibility to transform the product. This claims leads to the main hypothesis tested in this article:

Hypothesis 1 The functional models of products that are optimal have transformation costs that are lower than non-optimal products.

In this hypothesis, optimality is construed in a functional architecture sense, that is, products that have the minimum number of functions necessary to achieve an intended purpose, the minimum number of flows of energy, material, and information between the functions, and a structure that imposes the least constraints on transformation.

If certain functional architecture structures have a lower cost of transformation than others, it would be useful to know those structures a priori. The discussion in Sect. 2.2 about tree and small-world structures as being possible parsimonious structures leads to the following hypothesis:

Hypothesis 2 Products having functional models with tree or small-world structures will have lower transformation costs than products with other functional model structures.

It is predicted that the biological systems and their analogously design-engineered counterparts in the dataset should have a lower transformation cost between their current structure and target designs. Therefore, Hypothesis 1 will be tested by investigating whether the transformation cost for biological systems is lower than engineered products. Hypothesis 2 will be investigated by testing whether the transformation cost of products having a tree or small-world structure functional model is lower than others. According to Hypothesis 2, these tree or small-world structure products should have a lower transformation cost than their non-parsimonious structure counterparts. The following sections will test Hypothesis 2 by comparing the biological systems to the engineered products and tree or small-world structure functional models of both biological systems and engineered products to their non-tree and non-small-world counterparts.

4 Research method

4.1 Modeling products

The knowledge structure of a product is modeled according to a functional model and the ontology of the functional basis (Stone and Wood 2000). The functional model in turn is represented as a unipartite, directed graph $\mathcal{G} = (V, E, \mu, \nu)$ where V represents the nodes (functions), E represents the edges (flows) and $E \subseteq V \times V$. $\mu \rightarrow V$ labels the function nodes and $\nu \rightarrow E$ labels the edges. In the functional basis, flow types include energy, matter, and information. A unipartite graph was chosen because of the much smaller number of distinct edge types (flows) compared to function types in a typical product. The bipartite representation would bias the graphs toward star-like structures. The models produced may include loops and multiple edges between two nodes because different types of flows can exist between two nodes or the same flow between two nodes recur in the functional model. A decision was taken to represent each instantiation of a function as one, common function node rather than as separate function nodes each with a distinct label μ . This creates functional models with a fixed upper bound on the number of nodes. While more complex graph representations are possible in which distinct nodes and edges can be assigned unique identifiers (Wyatt et al. 2013), maintaining a fixed upper bound on the number of nodes eliminates the need to test the obvious question that functional models with more

nodes cost more to transform than models with fewer nodes.

Functional models were sourced from various published research articles (Nagel et al. 2008a, b; Nagel 2013; Weaver et al. 2011) and products in the Oregon State Design Repository (Stone 2014).⁴ The author aimed to identify as many as possible peer-reviewed published functional models represented according to the functional basis without discarding any models found unless they consisted of only a few functions and flows. In total, 30 products were identified and analyzed. While some functional models were derived from actual products and others from research into the potential design of a product, this is not a limitation in the dataset. The purpose of the research is to discover whether significant topological differences exist across the functional models. Models based upon research into the potential design may exclude some functions and flows for simplicity, but the overall functional model is likely to capture the essential functions and flows of the product. More importantly, it captures the *structure* of the functional architecture. Ten of the products are biological systems or biologically inspired engineered products having nearly the same functional model as their biological counterpart. These biological products are:

1. Abscission, armadillo armor, automobile airbag, housefly, Lexus SC430 convertible.
2. Microassembly abscission, puffer fish, retractable stadium roof (Nagel et al. 2008a).
3. Chemical sensor (Nagel 2013).
4. Sparrow flapping wing microair vehicle (FWMAV) (Bejgerowski et al. 2009; Nagel et al. 2008a).

The 20 engineered products are:

1. Brother sewing machine, crest toothbrush, drill, DeWALT sander, genie garage door opener, delta jigsaw, delta nail gun, juice extractor, lawn mower, mixer, seiko kinetic watch, solar yardlight, stapler, supermax hair dryer, vise (Stone 2014).
2. Digger the Dog (Nagel et al. 2008b).
3. Enviro energies wind tunnel, piezoelectric shoe heel impact harvester (Weaver et al. 2011; Weaver 2008).
4. Rice cooker, shop-vac vacuum cleaner (Caldwell et al. 2012).

To ensure consistency in the level of hierarchy in the functional models, each functional model was verified for descriptions at least at the second level for all functions or flows (Stone and Wood 2000) since the second level has

been shown to be the most useful and informative (Sen et al. 2010). However, it became necessary to refine some models to the third level of description of the functional basis. The third level was sometimes necessary to disambiguate a function or flow when the second level disregards important distinctions. For example, the electric toothbrush (Stone 2014) converts (function) energy–mechanical–rotational (flow) to energy–mechanical–translational (flow). The electric drill converts (function) energy–electrical into energy–mechanical–rotational (flow) and converts (function) energy–human into energy–mechanical–translational. Without the third level in the functional basis hierarchy for energy, the functional model would be ambiguous in relation to what form of mechanical energy is being converted. Some functional models must describe a product at the third level to ensure a physically correct description. For example, the housefly (Nagel et al. 2008a) can detect (function) a solid–liquid–mixture (material) and the puffer fish can export (function) a liquid–gas–mixture (material). As such, if a function or flow must be described at the third level for any product in the dataset, all other models were similarly described and hence modified at the third level for the particular function or flow.⁵ This is necessary to ensure commensurate transformation cost calculations for node or edge substitution (as described below). Where it is not necessary to model down to the third level, products were modeled at the second level of the functional basis as explained previously. The third-level descriptor *generic complements* was never used. Functional models were entered into Microsoft® Excel® and then converted into GraphML format using yEd (<http://www.yworks.com/en/products/yfiles/yed/>). The GraphML graphs were then imported into Mathematica® 10 for analysis.

To contextualize the functional models, some proposed measures of a design's structural complexity (Summers and Shah 2010; Mathieson and Summers 2010), the number of nodes and edges and the characteristic path length, are calculated. The characteristic path length is defined as the average number of edges in the shortest paths between all node pairs. In addition to custom code, graph analyses were performed in Mathematica® 10 using the following built-in functions: *GlobalClusteringCoefficient* for the clustering coefficient; *MeanClusteringCoefficient* for the Watts–Strogatz local clustering coefficient; and *GraphDistanceMatrix* to calculate the shortest paths between all node pairs using Dijkstra's method. The characteristic path length is the mean of the shortest paths.

⁴ One of the challenges in performing this research is the lack of a large number of functional models of complex products. Functional models were taken from existing data sources to limit potential bias in their production by the author.

⁵ The exception to this rule is the signal–status in parts of the functional models for Digger the Dog and abscission. None of the third-level descriptors correctly describe the type of signal. The second-level descriptor of signal–status was therefore used. This is not expected to alter the results significantly.

The SWI for each product was calculated by comparing its functional model graph to 1000 random small-world equivalent graphs.

4.2 Modeling transformation

The calculation of the transformation cost of a functional model entails two elements: first, identifying the minimal set of operations that can transform a functional model from the source to the target functional model; and second, constructing the target functional model. The target functional model is the new model to which the source model is transformed.

At least two methods of calculating the transformation cost are possible once a functional model is represented as a graph. The GED [see Gao et al. (2010) for a review of GED algorithms] is the sequence of operations that transform a source graph into a target graph. A GED algorithm calculates the minimal set of operations that can transform a graph representation from a source model to a target model. Second, researchers in computer science have proposed graph kernels to measure the distance of architectural change between versions of software (Nakamura and Basili 2005). The graph kernels measure the similarity between two software architectures; the overlapping similarity defines the kernel of the software architecture. The main advantage of the GED approach is that the method computes various sequences of transformation and selects the least cost sequence, which is defined as the GED, from this set. A sequence need not be unique. While this study does not yet make use of the sequence of transformation, it is anticipated that sequence information will be relevant to physical systems architecture design. For example, a company may prefer a function transformation sequence that coincides with the decision order through which system-level parameters flow down to component-level design parameters (Eckert et al. 2012).

Graph edit transformations consist of three operators: node insertion or deletion; edge insertion or deletion; and node or edge substitution. Figure 1 illustrates a source graph which should be transformed into the target graph of Fig. 2. One possible sequence of transformation operations is:

1. Insert node import and its associated edge.
2. Delete node transport and its associated edges.
3. Substitute node shape with node store.

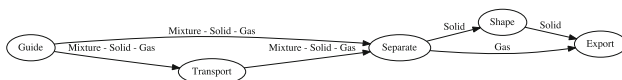


Fig. 1 Source graph

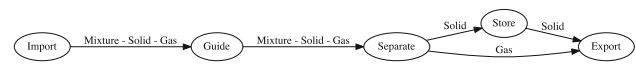


Fig. 2 Target graph

A cost function is defined for each operation. The total transformation cost is the sum of all costs for operations in the sequence. A number of feasible sequences are possible to transform a source graph into the target graph. The lowest cost transformation is reported as the GED.

The numerical value of the cost functions measures the amount of transformation required by each edit operation. The cost functions can be arbitrarily set to represent the expense of the structural transformation and should monotonically increase over the given distance metric. For example, one could set the cost of transformation by knowledge substitution (node substitution) higher than knowledge addition (node addition) since substitution may have cascading effects. In practice, the cost of adding, deleting, or substituting a function could be proxied by empirical values for R&D expenditures. Determining these values is beyond the scope of this article, but could form the basis of valuable empirical research.

While it is possible to calculate the exact GED for relatively small graphs, the problem grows exponentially with the number of nodes. That is, there is an exponential growth in the possible number of sequences. Given the relatively small number of nodes in this research, an A* algorithm (Hart et al. 1968) (“Appendix”) implemented in a graph matching toolkit (Riesen et al. 2013) was used to calculate the GED. In the absence of empirical data on the cost of knowledge addition, deletion, and substitution, the cost of a node or edge addition or deletion was set as the multiplication of the cost of the operation (set to 1). The string edit distance between the current label and the target label for the node or edge is used to calculate the cost of substitution. The label describes the function or the flow. The calculation of the string edit distance is based upon the Levenshtein distance. For example, the Levenshtein string edit distance between the flows energy—biological and energy—electromagnetic is 13. To test a large number of target product models to which a functional model could be transformed, thereby simulating possible design innovations that a human design engineer could introduce and to which a product could be transformed, 1000 equivalent random, target functional models were constructed for each product. An equivalent functional model has the same number of nodes and node degree as the source functional model, but not necessarily the same functions or edges as the source functional model. That is, which nodes are connected and the labels for the nodes and edges will differ between the source and target functional models. Further, three types of graph structures for the target functional models were generated: tree, small-world, and Bernoulli

random. In all, 3000 random equivalent target functional models are generated per source functional model, thereby providing a very broad comparison to possible functional architectures that might be able to achieve the same purpose as the original source functional model.

According to the discussion in Sect. 2.2, tree or small-world networks are parsimonious functional model structures. The GED from these structures provides an indication of the sub-optimality of the source functional model. A Bernoulli random graph is a baseline graph structure, that is, a graph that represents an arbitrary functional model to which the product could be transformed.

In the following explanations of graph construction methods, the rewiring probability p is calculated as the ratio of actual to total possible edges in a given functional model. For example, if a functional model has v edges and n nodes, then the rewiring probability is given by $\frac{v}{n(n-1)}$. The random graphs were constructed using built-in graph construction functions in Mathematica[®] 10: *RandomGraph* with *WattsStrogatzGraphDistribution* for small-world graphs; *RandomGraph* with *BernoulliGraphDistribution* for Bernoulli random graphs; and *RandomGraph* with n nodes and $n - 1$ edges for tree-like graphs. A small-world graph is constructed by starting with a regular graph and rewiring each edge by changing one of the nodes with probability p such that no loop or multiple edge is created. A Bernoulli random graph is generated by first creating a complete graph with n nodes. Edges are retained with a probability p according to a Bernoulli distribution. A random tree-like graph is constructed by starting with an n nodes and adding an edge according to a uniform distribution until $n - 1$ edges have been added. Node and edge labels for the random graphs are assigned randomly by selecting a label from the list of node or edge labels in the source functional model, respectively. While the random selection of edge labels may lead to functional models that are not physically possible, the aim is to generate a large number of samples of arbitrary designs to which the source product could be transformed as a way to test the source functional models' constraints on transformation. The repetition of edge types in the functional models increased the likelihood of selection of common flows in a product. All graphs constructed by Mathematica were exported in GXL format to calculate the GED using the Java-based graph matching toolkit (Riesen et al. 2013).

5 Results

Table 1 reports the graph properties of the products analyzed including topological properties for the structural complexity of designs (Summers and Shah 2010;

Mathieson and Summers 2010). The products have been categorized into the 'Biological' group if the model relates to a biological system or its bio-inspired engineered analog or the 'Engineered' group otherwise. Nearly all of the products in the Biological Group have a 0 clustering coefficient even though their node degrees (in and out) are similar to the values for products in the Engineered Group. Distribution tests of the SWI results confirmed that they do not follow a normal distribution ($p < .001$). Figures 3 and 4 show the skew in the SWI distribution and poor fit to the calculated normal probability distribution function for two products, the Genie[®] garage door opener and Digger the Dog[™], respectively. Therefore, median values of SWI are reported in Table 1.

A visual analysis of the functional models rendered using Mathematica's internal graph drawing tools suggests that the functional models are of two types: tree (or star-like) and non-tree-like. The graph illustrations of Figs. 5 and 6 typify this difference in the structure of the functional models. The abscission (Nagel et al. 2008a) functional model in Fig. 5 appears tree-like, having no transitive loops and therefore a clustering coefficient (local and global) of 0. The drill (Stone 2014) functional model in Fig. 6, in contrast, has a non-tree-like structure and likewise a nonzero clustering coefficient. The following results test the hypotheses to determine whether these structural differences occur across all of the products tested and whether they produce differences in the GED.

The GED for each of the products was calculated as the distance between the current functional model and its small-world, Bernoulli random, or tree-like equivalent.⁶ The results of the GED computations are shown in Table 2. Median values are reported because a Shapiro–Wilk test of normality showed that none of the GED distributions were normal ($p < .05$). Since the data do not follow a normal distribution, the Kruskal–Wallis test is used to compare the GEDs between the groups.

Hypothesis 1 states that the transformation cost is lower for optimal products. This hypothesis is tested by comparing the GED between the Biological and Engineered Groups. First, the hypothesis is tested by comparing the median values of the GED as reported in Table 2. The median value test considers variance in GED wherein the artificiality of the product is the source of variance. It can be concluded that the median GED in the Biological Group ($n_1 = 10$) is statistically significantly lower than in the Engineered Group ($n_2 = 20$), as shown in Table 3. Thus, nature's designs can be said to be more flexible to change than engineered products. The second test compares the

⁶ For 30 product models, 1000 GED analyses per product required about 3 days of compute time on a dual-CPU Dell Precision T7500 with 16 GB of RAM.

Table 1 Graph properties of products

| | Nodes | Edges | CC-WS | CC | In-degree | Out-degree | CPL | SWI |
|--|-------|-------|--------|--------|-----------|------------|---------|----------|
| <i>Biological Group</i> | | | | | | | | |
| Abscission | 7 | 24 | 0 | 0 | 1.71 | 1.71 | 1.85 | 0 |
| Armadillo armor | 10 | 26 | 0 | 0 | 1.30 | 1.30 | 3.11111 | 0 |
| Automobile air bag | 12 | 30 | 0 | 0 | 1.25 | 1.25 | 4.71429 | 0 |
| Chemical sensor | 9 | 22 | 0 | 0 | 1.22 | 1.22 | 3.19355 | 0 |
| Housefly | 11 | 34 | 0 | 0 | 1.55 | 1.55 | 14.4835 | 0 |
| Lexus® SC430 convertible roof | 12 | 26 | 0 | 0 | 1.08 | 1.08 | 2.06667 | 0 |
| Microassembly abscission | 8 | 24 | 0 | 0 | 1.50 | 1.50 | 2.44 | 0 |
| Puffer fish | 12 | 32 | 0 | 0 | 1.33 | 1.33 | 3.45455 | 0 |
| Retractable stadium roof | 10 | 26 | 0 | 0 | 1.30 | 1.30 | 3.11111 | 0 |
| Sparrow FWMAV | 8 | 18 | 0.271 | 0.273 | 1.13 | 1.13 | 2.31034 | 0.990722 |
| <i>Engineered Group</i> | | | | | | | | |
| Bench vise | 5 | 16 | 0 | 0 | 1.60 | 1.60 | 1.63636 | 0 |
| Brother® sewing machine | 11 | 70 | 0.279 | 0.350 | 3.18 | 3.18 | 2.18072 | 1.8205 |
| Crest® toothbrush | 10 | 28 | 0.145 | 0.200 | 1.40 | 1.40 | 2.14634 | 0.786823 |
| Delta® jigsaw | 14 | 34 | 0 | 0 | 1.21 | 1.21 | 3.39063 | 0 |
| Delta® nail gun | 9 | 18 | 0 | 0 | 1.00 | 1.00 | 4.26667 | 0 |
| DeWALT® sander | 12 | 48 | 0.448 | 0.308 | 2.00 | 2.00 | 3.07843 | 2.29857 |
| Digger the Dog™ | 13 | 44 | 0.256 | 0.214 | 1.69 | 1.69 | 6.39394 | 0.674474 |
| Drill | 15 | 56 | 0.0456 | 0.0732 | 1.87 | 1.87 | 4.225 | 0.215213 |
| Enviro-Energies® wind tunnel | 15 | 42 | 0.117 | 0.0909 | 1.40 | 1.40 | 4.15652 | 0.517549 |
| Genie® garage door opener | 11 | 52 | 0.357 | 0.214 | 2.36 | 2.36 | 2.17105 | 2.40102 |
| Juice extractor | 12 | 34 | 0 | 0 | 1.42 | 1.42 | 3.88889 | 0 |
| Lawn mower | 12 | 44 | 0 | 0 | 1.83 | 1.83 | 2.68 | 0 |
| Mixer | 10 | 32 | 0.150 | 0.130 | 1.60 | 1.60 | 2.59091 | 0.701912 |
| Piezoelectric shoe heel impact harvester | 17 | 50 | 0.0213 | 0.0732 | 1.47 | 1.47 | 3.83523 | 0.113865 |
| Rice cooker | 9 | 30 | 0.176 | 0.167 | 1.67 | 1.67 | 3.64706 | 0.481692 |
| Seiko® kinetic watch | 16 | 50 | 0.100 | 0.231 | 1.56 | 1.56 | 4.10294 | 0.461636 |
| Shop Vac® | 10 | 34 | 0.117 | 0.150 | 1.70 | 1.70 | 3.0 | 0.452423 |
| Solar yard light | 5 | 10 | 0.450 | 0.500 | 1.00 | 1.00 | 1.76923 | 0.719441 |
| Stapler | 12 | 32 | 0 | 0 | 1.33 | 1.33 | 3.34568 | 0 |
| Supermax® hair dryer | 9 | 30 | 0 | 0 | 1.67 | 1.67 | 3.34783 | 0 |

CC-WS Watts–Strogatz local clustering coefficient, CC global clustering coefficient, CPL characteristic path length, SWI small-world index

individual values of GED between products in the biological ($n_1 = 10,000$) and engineered ($n_2 = 20,000$) groups. This statistical test takes into consideration variance in GED due to the artificiality of the product (engineered or biological) and in the range of designs to which a product could be transformed. The conclusion is the same ($p < .001$). Figure 7 shows the distribution of median GED values, and Fig. 8 shows the distribution of all the GED values to support the preceding statistical results.

Hypothesis 2 states that products having tree or small-world functional model structures will have a lower transformation cost than products not having these structures. Nearly all of the products in the Biological Group,

except the sparrow FWMAV, had a local and global clustering coefficient of 0. A signed rank test of the SWI for the sparrow FWMAV rejects the null hypothesis that the median SWI ≤ 1 ($p = .00126631$), confirming that its SWI satisfies the sufficient test for small-world networks (Humphries et al. 2006). Therefore, all of the biological systems have tree or small-world structures. According to the prior results, the items in the Biological Group have a lower GED than the items in the Engineered Group. It is therefore possible to confirm Hypothesis 2 when comparing biological systems to engineered products.

Hypothesis 2 is further tested by comparing the GED between products having tree or small-world functional

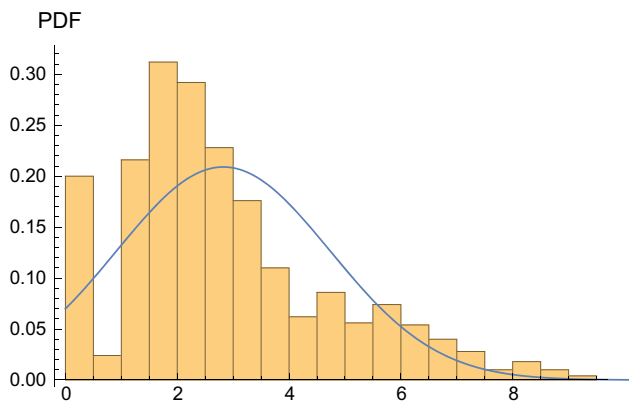


Fig. 3 Histogram and calculated probability distribution function of SWI values for Genie® opener

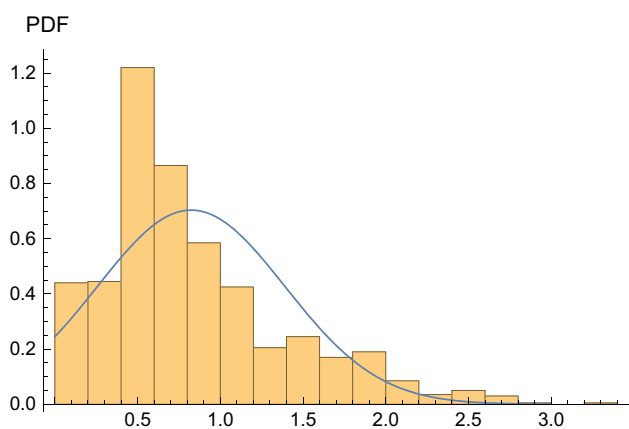


Fig. 4 Histogram and calculated probability distribution function of SWI values for Digger the Dog™

model structures and those without these structures. Seven products in the Engineered Group had a local and global clustering coefficient of 0. This implies that they have tree-like structures. Three engineered products satisfied the condition for a small-world network: the DeWALT sander, the Genie garage door opener, and the Brother sewing machine. A signed rank test of the SWI distributions for these three products rejected the null hypothesis that the median SWI ≤ 1 ($p < .001$), confirming that their SWI satisfies the sufficient test for small-world networks (Humphries et al. 2006). Figures 3 and 4 show two products having a SWI > 1 or < 1 , respectively, for some small-world graph equivalents, but a statistical tendency toward the results described. For all other engineered products having nonzero values of SWI, signed rank tests failed to reject the null hypothesis that the median value of SWI ≤ 1 ($p > .1$).

Based upon the discussion in Sect. 2 regarding parsimonious functional model structures for products, this set of ten engineered products may be thought of as having parsimonious knowledge structures. If these 10 engineered

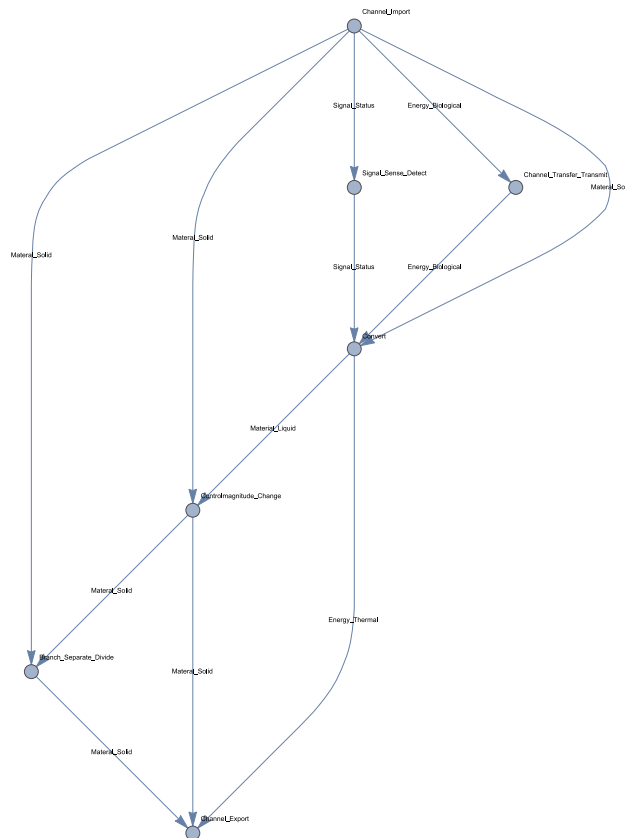


Fig. 5 Tree-like structure of abscission (Nagel et al. 2008a)

products ($n_1 = 10$) have parsimonious knowledge structures, then, according to Hypothesis 2, their transformation costs should be lower than the other engineered ($n_2 = 10$) ones. It is not possible to reject the null hypothesis of Hypothesis 1 between the tree or small-world structure engineered products and the other engineered products (small-world $p = .496$, Bernoulli, $p = .910$, tree-like $p = .762$) when comparing the median values of GED. The same statistical test comparing individual values of GED between tree or small-world structure engineered products ($n_1 = 10,000$) and non-tree or small-world structure engineered products ($n_2 = 10,000$) confirms ($p < .001$) that tree or small-world structure engineered products have a lower GED (small-world mean rank = 9298.11; Bernoulli mean rank = 9781.51; tree-like mean rank = 9593.11) (small-world mean rank = 10,702.89; Bernoulli mean rank = 10,219.49; tree-like mean rank = 10,407.89) than non-tree or small-world structure engineered products. Thus, Hypothesis 2 can be confirmed for the engineered products when random variation in the target designs is considered, but the statistical test lacks sufficient power when the source of variance is the type of engineered product, e.g., solar yard light or juice extractor. Finally, the tree or small-world structure engineered products are compared against the Biological Group. Since all of the



Fig. 6 Graph structure of a drill (Stone 2014)

items in this set have tree or small-world structure structures, the null hypothesis of Hypothesis 2 should not be rejected. They should all have similarly low GEDs. It is not possible to reject the null hypothesis of Hypothesis 1 between the tree or small-world structure engineered products and the Biological Group (small-world $p = .075$, Bernoulli, $p = .052$, tree-like $p = .052$) when comparing the median values of GED. However, since the p values are approaching .05, it is possible that with more samples, it would be confirmed that products in the Biological Group have a lower GED than tree or small-world structure engineered products. The mean ranks of the Biological Group (small-world mean rank = 8.15; Bernoulli mean rank = 7.95; tree-like mean rank = 7.95) are lower than the mean ranks of the Engineered Group (small-world mean rank = 12.85; Bernoulli mean rank = 13.05; tree-like mean rank = 13.05). However, the individual GEDs of the Biological Group (small-world mean rank = 8001.43; Bernoulli mean rank = 7745.41; tree-like mean rank = 7753.66) are lower than the GED for the optimal engineered products (small-world mean rank = 11,999.57; Bernoulli mean rank = 12,255.29; tree-like mean rank = 12,247.34) for all types of target graphs ($p < .001$). Since the GED in the Biological Group is lower than even the tree or small-world structure products in the Engineered Group, the structure of the functional models in the Biological Group is the most flexible to change in the analyzed dataset. These results state that having a tree or

small-world structure is a sufficient basis to judge a knowledge structure as being optimal only when comparing engineered products, and possibly only when comparing products within a particular class. There are characteristics of products in the Biological Group that make their GED lower than products in the Engineered Group. The next section discusses some of these characteristics and the ensuing design guidelines.

Overall, the results confirm Hypotheses 1 and 2. The GED of the Biological Group is lower than the GED of the Engineered Group. It was confirmed that all items in the Biological Group had a tree or small-world structure. The GED of tree or small-world structure products in the Engineered Group is lower than the GED of non-tree or small-world structure engineered products. These results suggest that transformation cost is an important criterion of the optimality of a product.

6 Discussion

6.1 Summary

The premise of this paper is that the underlying knowledge structure of designs has real consequences, which include its technological progress potential (Dong and Sarkar 2015) and, as studied in this paper, its potential for change by transformation. The hypothesis tested in this

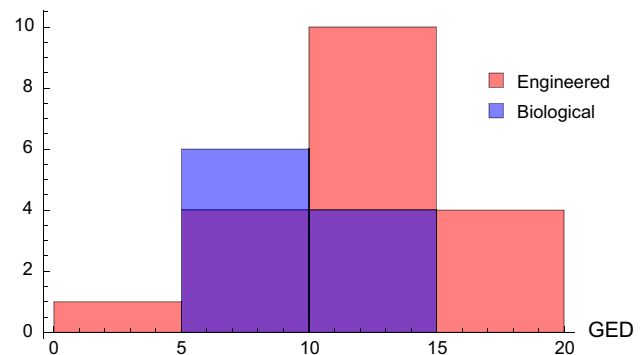
Table 2 Graph edit distance between source model and target graphs having the given structural properties, small-world, Bernoulli random, or tree-like

| Model | Small-world | Bernoulli | Tree-like |
|--|-------------|-----------|-----------|
| <i>Biological Group</i> | | | |
| Abcission | 7.25 | 6.75 | 5 |
| Armadillo armor | 9.25 | 6.75 | 5.75 |
| Automobile air bag | 11.25 | 8 | 7 |
| Chemical sensor | 8.25 | 6 | 5.25 |
| Housefly | 10.75 | 9.25 | 7.25 |
| Lexus® SC430 convertible roof | 10.25 | 6.75 | 6.25 |
| Microassembly abcission | 8 | 6.75 | 5.25 |
| Puffer fish | 11.25 | 8.5 | 7 |
| Retractable stadium roof | 9.25 | 7 | 5.75 |
| Sparrow FWMAV | 7 | 4.75 | 4.25 |
| <i>Engineered Group</i> | | | |
| Bench vise | 5.75 | 4.75 | 3.75 |
| Brother® sewing machine | 15.25 | 18.75 | 12 |
| Crest® toothbrush | 9.5 | 7.75 | 6.25 |
| Delta® jigsaw | 12.75 | 9 | 7.75 |
| Delta® nail gun | 7.5 | 4.75 | 4.5 |
| DeWALT® sander | 13.75 | 13.5 | 9.75 |
| Digger the Dog™ | 13.25 | 11.75 | 9.25 |
| Drill | 16 | 15 | 11.25 |
| Enviro-energies® wind tunnel | 14.5 | 11 | 9.25 |
| Genie® garage door opener | 13.5 | 14.5 | 9.75 |
| Juice extractor | 11.75 | 9.25 | 7.5 |
| Lawn mower | 13 | 12 | 9 |
| Mixer | 10 | 8.75 | 6.75 |
| Piezoelectric shoe heel impact harvester | 16.75 | 13 | 10.75 |
| Rice cooker | 9.5 | 8.75 | 6.5 |
| Seiko® kinetic watch | 16.25 | 13.5 | 10.5 |
| Shop Vac® | 11 | 9.75 | 7.5 |
| Solar yard light | 3.75 | 2.5 | 2.25 |
| Stapler | 10.25 | 8.25 | 7 |
| Supermax® hair dryer | 9.25 | 8.25 | 6.25 |

Table 3 Nonparametric tests of differences between medians of graph edit distances

| Structure | Group | Mean rank | <i>p</i> value | χ^2 |
|-------------|------------|-----------|----------------|----------|
| Small-world | Biological | 10.55 | .029 | 4.750 |
| | Engineered | 17.98 | – | – |
| Bernoulli | Biological | 9.45 | .008 | 7.113 |
| | Engineered | 18.53 | – | – |
| Tree-like | Biological | 9.90 | .014 | 6.087 |
| | Engineered | 18.30 | – | – |

paper is that the structure of the functional model of a product can lock-in a design by influencing the cost of transformation. The article operationalizes the cost of transformation using the GED. By comparing the GEDs of biological systems and their analogously designed

**Fig. 7** Distribution of median GED values

engineered products, which are assumed to be optimal, to those of biological systems, the results confirmed that the biological systems had a statistically significantly

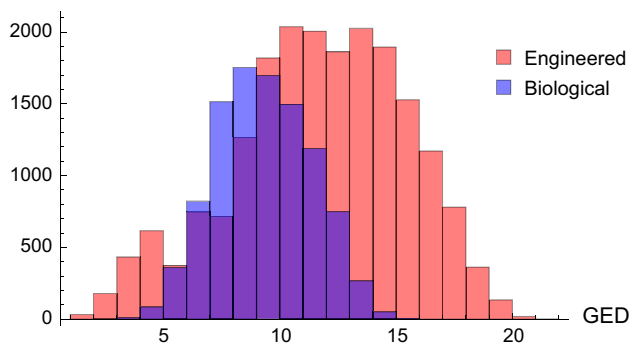


Fig. 8 Distribution of GED values

lower GED than the engineered ones. engineered products having tree or small-world functional model structures had a lower GED than non-tree or small-world engineered products. These results imply that even though two products can achieve the same intended purpose, one can be more flexible than the other in relation to design innovation by transformation due to differences in the structure of their underlying functional models. The structure locks-in function and flow choices and the resulting component choices that are possible. Based upon these results, a tentative hypothesis is that products having tree-like or small-world-like functional architectures will progress faster than those with other functional architecture topologies.

For the engineered products that had tree or small-world functional models, that their corresponding GEDs were nonzero and higher than the median suggests that even products having parsimonious knowledge structures have some inbuilt inefficiencies. An examination of these functional models showed that inefficiencies are caused by transitive loops between various functions, as these loops do not appear in the Biological Group. Given the relation between increasing node and edge count and increasing GED, the results also imply that engineers should aim to reduce the total number of distinct functions and flows to the minimally necessary and sufficient set.

The comparison between the functional models in the Engineered Group and the Biological Group provides design principles to simplify functional models. The design principles are:

1. Eliminate transitive loops between various functions (e.g., use an open-loop controller rather than a closed-loop controller where possible⁷).
2. Minimize the total number of distinct functions or flows by:

- (a) Combining n functions or flows into $n - m$ functions or flows where $1 \leq m < n$ (e.g., locating pins guide a work piece and combine the functions of couple and stabilize).
- (b) Eliminating a function or flow (e.g., electric motors, as compared to internal combustion engines, can produce full torque over a wide range of operating speed, thereby eliminating some transmission elements; additive manufacturing is based upon the principle of eliminating functions associated with the removal of material from a work product).

3. Minimize repetition of functions and flows.

The latter two design principles should be expected: Economy in function and flow is highly desirable.

The first design principle illustrates inefficiencies built into engineered products. As other researchers have already illustrated about biological products (Vincent et al. 2006), variety in function is produced by hierarchical and recursive manipulations of materials at larger sizes. In engineered systems, variety in function is achieved by the replication of functions and the instantiation of new functions. In so doing, transitive loops are produced, as flows of energy and material are passed to another component that repeats a prior function. Transitive loops have the following form: edges $1 \rightarrow 2$ and $2 \rightarrow 3$ transitively connect $1 \rightarrow 3$. Designers should, if possible, transform the structure such that unnecessary relations are eliminated by the transitive loop. For example, suppose the function structure contains the transitive loop, evident in the Convert \rightarrow Controlmagnitude_Actuate \rightarrow Controlmagnitude_Regulate \rightarrow Convert flow of energy—mechanical—translational loop of Fig. 6. The desired structure could eliminate the Controlmagnitude_Regulate \rightarrow Convert edge by using an open-loop controller, if appropriate.

Simpler functional architectures should lead to less complex component configurations (Mathieson et al. 2011) and improved upgradeability (Umeda et al. 2005). The complexity of a functional architecture is not the only determinant of the complexity of the resulting physical architecture though. Performance requirements also influence the physical complexity (Peterson et al. 2012). These results imply as a design policy that designers should strive to reduce the complexity of the underlying functional architecture for a product because the functional architecture will impose a cost on future design changes.

6.2 Further applications

The method presented in this paper aims to discover the properties of the structure of functional models as a way to quantify their transformability into another structure. The

⁷ An open-loop controller is simpler to implement than a closed-loop controller since it does not require a feedback mechanism, i.e., no control signal flow and no transitive loops of energy flow between functions such as control magnitude and channel.

method of GED, though, could be generalized to quantify the difference (or similarity) between the structures of functional models. In this sense, this paper shares the same line of thinking as research in discovering the structure of large repositories of designed objects as a way to transfer knowledge between designs having similar underlying structures (Fu et al. 2013). While at the moment it is not known whether engineered objects and systems might have regular patterns, understanding the consequences of the knowledge structures of designs, that is, their genotype structure, rather than their physical architectures, their phenotype, may yield new techniques to optimize designs starting from their functional description.

In addition, the method of calculating the GED provides an objective, knowledge-based method to measure the difference between designs at the level of their functional model. This has application in the problem of measuring the creativity of designs. Creativity is normally assessed by several factors but novelty relative to other designs in its ‘class’ is a recurrent criterion. The GED can be applied to measure the degree of transformation necessary to modify the functional model of one design to another; novel designs are further away from the others because they have different underlying knowledge structures. In addition, the GED between a proposed design’s functional model and an optimal equivalent can yield one other important and underlooked creativity metric: economy.

6.3 Limitations and improvements

Given the dataset, the empirical analyses should be characterized as exploratory since the primary focus in this paper is to introduce and begin to develop a new and potentially impactful concept of functional lock-in based on the transformation cost of functional models of products. The important question is whether the results occur due to a bias in the dataset rather than due to intrinsic differences between biological systems and engineered products. There are good reasons to believe that the results do not occur solely due to the choice of products (the dataset itself). Firstly, the dataset was obtained through a careful literature review of peer-reviewed, published research in which authors have provided a functional model according to the functional basis. Given a bias to publish positive results, it is likely that those functional models represent ‘best-in-class’ products. As a consequence, they may not represent the true population of biological systems or engineered products. Nonetheless, as long as the products in both sets are ‘best-in-class’, then the comparison between the sets remains fair. It is useful to note that many of the engineered products have been in production for a very long period of time; for example, the sewing machine was invented in the sixteenth century. It is

therefore likely that the engineered products have had sufficient time to be significantly improved, if not optimized, making them a useful comparison to the biological systems. Secondly, a very large number of random equivalent target models (potential functional models for the products) were generated to sample a broad transformation space and thereby obtain a median GED, rather than a single point value. Finally, the functional models in both sets are not too dissimilar in size. As shown in Table 1, the products in both sets have a similar number of nodes/functions (two-sample t test $p = .1977$) but the engineered products have more edges/flows (two-sample t test $p = .0188$). However, the optimal engineered products did not have significantly fewer nodes ($p = .1890$) or edges ($p = .5118$) than the non-optimal engineered products. Therefore, the slight difference in number of edges in the engineered products can only partially explain the results. What would be true is that had more complicated engineered products been included, such that there were an order of magnitude more edges, then the results would have certainly been skewed against the engineered products. As stated previously, the number of nodes in any graph is restricted to the maximum number of unique functions in the functional basis. Therefore, the patterns observed are not simple consequences of a sampling bias introduced by using functional models arising from different databases.

Ideally, it would be preferable to study the transformation of a single product for which the intended purpose has not changed over a long period of time, but the functional model has changed. An example of such a product is a mass data storage device. These devices have transformed from paper-based punch cards to optical disks to transistor-based solid-state memory chips. Examining the rate of improvement *within* each generation of device and *across* each generation of device and their correlation to the GED of the corresponding devices would provide further empirical data on the constraints imposed by a knowledge structure on the rate of improvement of a product. As well, a more sophisticated design grammar could be applied to describe the engineering changes made to these devices over time.

An important limitation in the method for calculating the GED is that the cost of node and edge operations is set to unity multiplied by a distance metric based upon the Levenshtein edit distance between the source and target labels of the nodes and edges. The true cost should be established through empirical data on the actual cost of adding, deleting, and substituting functions and flows. Such data should be available from R&D data and would form a substantial part of empirical research in forecasting the long-term trajectory and costs of design innovation by transformation.

The grammar for the transformation operations could be improved. In the present model, transformations consist of

random node insertion or deletion, edge insertion or deletion, and node or edge substitution. Instead, a design grammar could be defined to operate at the level of functions (nodes) and flows (edges). At the moment, most design grammars operate directly at the level of the physical instantiation of the product. The grammar may either use shapes, i.e., shape grammars, or symbols to define elements and operations which, when recursively applied, lead to a more complex, compound element. In the situation of a functional model, a grammar would define the appropriate set of transformations to nodes (functions) and edges (flows), while respecting appropriate physics-based constraints. For example, it would be possible to substitute the nodes and edges associated with importing energy–human channel–import (energy–human) channel–guide (energy–human) support–secure (energy–human) convert (energy–mechanical–translational) with the nodes and edges import (energy–electrical) channel–transfer–transmit (energy–electrical) convert (energy–mechanical–translational). Certain edge substitutions would not be possible, such as substituting a flow of energy–electrical with a flow of energy–hydraulic. In addition, each of these transformations could be assigned a different cost based upon empirical data on the type and cost of the components associated with the transformations. Developing the design grammar for the allowable transformations and their cost of implementation would present important empirical research in elaborating upon this fundamental model.

7 Conclusion

This paper claims that the structure of a product’s knowledge constrains the ability to transform the design. The functional architecture locks-in commitments to a set of functions and flows due to the structure of the connections between functions and flows. To test the claim, the paper compared the GEDs of biological systems and engineered products, on the assumption that biological systems were optimal. The results conclude that the GED of optimal products is lower than their non-optimal counterparts. The paper further compared engineered products having parsimonious functional architecture structures to their non-parsimonious engineered counterparts and concluded that the GEDs for the engineered products having parsimonious functional architecture structures were lower. The engineered products having tree or small-world functional model structures could therefore be described as having a more optimal functional architecture than the other engineered products. The results point to the general claim that the cost of transformation of optimal products is lower than their non-optimal counterparts and the functional lock-in is weaker. This article begins to illustrate how the structure of

knowledge underlying a product can constrain its rate of improvement, and thereby serve as a way to forecast potential rates of improvement. This constraint occurs regardless of the physical architecture. The implication is that a fully integrated system architecture in which the underlying functional architecture is separable is preferable to a modularized system in which there is significant overlap in functional architecture between the sub-systems. The fundamental hypothesis is that as the topology of the functional architecture of a product approaches optimality, the cost of design innovation by transformation decreases. Further empirical validation of this hypothesis would provide a powerful basis for both the redesign of engineered products and the estimation of rates of improvement. More broadly, the paper proposes minimum transformation cost as a property of an optimal design.

Acknowledgments Andy Dong is a recipient of an Australian Research Council Future Fellowship (FT100100376). The author acknowledges the set of functional models provided by Professor Joshua Summers of Clemson University.

Appendix: A* algorithm for graph edit distance

Algorithm 1 A* ALGORITHM TO COMPUTE MINIMUM EDIT DISTANCE [24]

```

1: Input: Non-empty graphs  $g_1 = (V_1, E_1, \mu_1, \nu_1)$  and  $g_2 = (V_2, E_2, \mu_2, \nu_2)$  where  $V_1 = \{u_1, \dots, u_{|V_1|}\}$  and  $V_2 = \{u_2, \dots, u_{|V_2|}\}$ 
2: Output: Minimum cost edit path  $S$  from  $g_1$  to  $g_2$ 
3:
4:  $S \leftarrow null$ 
5:  $S \leftarrow$  Insert substitution  $\{u_1 \rightarrow w\}$  for each vertex  $w \in V_2$ 
6:  $S \leftarrow$  Insert deletion  $\{u_1 \rightarrow \epsilon\}$ 
7: loop
8:   Remove  $p_{min} = \arg \min_{p \in S} \{g(p) + h(p)\}$  from  $S$ 
9:   if  $p_{min}$  is a complete edit path then
10:     Return  $p_{min}$  as the solution
11:   else
12:     Let  $p_{min} = \{u_1 \rightarrow v_{i_1}, \dots, u_k \rightarrow v_{i_k}\}$ 
13:     if  $k < |V_1|$  then
14:       For each  $w \in V_2 \setminus \{v_{i_1}, \dots, v_{i_k}\}$ , insert  $p_{min} \cup \{u_{k+1} \rightarrow w\}$  into  $S$ 
15:       Insert  $p_{min} \cup \{u_{k+1} \rightarrow \epsilon\}$  into  $S$ 
16:     else
17:       Insert  $p_{min} \cup \bigcup_{w \in V_2 \setminus \{v_{i_1}, \dots, v_{i_k}\}} \{\epsilon \rightarrow w\}$ 
18:     end if
19:   end if
20: end loop

```

References

- Banbury CM, Mitchell W (1995) The effect of introducing important incremental innovations on market share and business survival. *Strategic Management Journal* 16(S1):161–182
- Bejgerowski W, Ananthanarayanan A, Mueller D, Gupta SK (2009) Integrated product and process design for a flapping wing drive

- mechanism. *Journal of Mechanical Design* 131(6):011010. doi:[10.1115/1.3116258](https://doi.org/10.1115/1.3116258)
- Braha D, Bar-Yam Y (2007) The statistical mechanics of complex product development: empirical and analytical results. *Manag Sci* 53(7):1127–1145
- Brusoni S, Prencipe A (2001) Unpacking the black box of modularity: technologies, products and organizations. *Ind Corp Change* 10(1):179–205
- Cai YL, Nee AYC, Lu WF (2009) Optimal design of hierarchic components platform under hybrid modular architecture. *Concur Eng* 17(4):267–277
- Caldwell BW, Thomas JE, Sen C, Mocko GM, Summers JD (2012) The effects of language and pruning on function structure interpretability. *J Mech Des* 134(6):061001
- Cardin MA, Kolfshoten G, Frey D, de Neufville R, de Weck O, Geltner D (2013) Empirical evaluation of procedures to generate flexibility in engineering systems and improve lifecycle performance. *Res Eng Des* 24(3):277–295
- Chiriac N, Hölttä-Otto K, Lysy D, Suh ES (2011) Level of modularity and different levels of system granularity. *J Mech Des* 133(10):101007–101010
- Clarkson PJ, Simons C, Eckert C (2004) Predicting change propagation in complex design. *J Mech Des* 126(5):788–797. doi:[10.1115/1.1765117](https://doi.org/10.1115/1.1765117)
- de Neufville R, Scholtes S (2011) *Flexibility in engineering design*. MIT Press, Cambridge
- Denman J, Sinha K, de Weck O (2011) Technology insertion in turbofan engine and assessment of architectural complexity. In: Eppinger SD, Maurer M, Eben K, Lindemann U (eds) 13th international design structure matrix (DSM) conference, Carl Hanser, Cambridge, MA, USA, pp 407–420
- Dong A, Sarkar S (2015) Forecasting technological progress potential based on the complexity of product knowledge. *Technol Forecast Soc Change* 90, Part B:599–610
- Dosi G (1982) Technological paradigms and technological trajectories: a suggested interpretation of the determinants and directions of technical change. *Res Policy* 11(3):147–162
- Eckert C, Keller R, Clarkson PJ (2011) Change prediction in innovative products to avoid emergency innovation. *Int J Technol Manag* 55:226–237
- Eckert CM, Stacey M, Wyatt D, Garthwaite P (2012) Change as little as possible: creativity in design by modification. *J Eng Des* 23(4):337–360
- Eisenbart B, Gericke K, Blessing L (2013) An analysis of functional modeling approaches across disciplines. *Artif Intell Eng Des Anal Manuf* 27:281–289. doi:[10.1017/S0890060413000280](https://doi.org/10.1017/S0890060413000280)
- Fei G, Gao J, Owodunni O, Tang X (2011) A method for engineering design change analysis using system modelling and knowledge management techniques. *Int J Comput Integr Manuf* 24(6):535–551. doi:[10.1080/0951192X.2011.562544](https://doi.org/10.1080/0951192X.2011.562544)
- Fernandes J, Henriques E, Silva A, Moss MA (2015) Requirements change in complex technical systems: an empirical study of root causes. *Res Eng Des* 26(1):37–55. doi:[10.1007/s00163-014-0183-7](https://doi.org/10.1007/s00163-014-0183-7)
- Fu K, Cagan J, Kotovsky K, Wood K (2013) Discovering structure in design databases through functional and surface based mapping. *J Mech Des* 135(3):031006
- Fujimoto T (2014) The long tail of the auto industry life cycle. *J Prod Innov Manag* 31(1):8–16
- Gao X, Xiao B, Tao D, Li X (2010) A survey of graph edit distance. *Pattern Anal Appl* 13:113–129. doi:[10.1007/s10044-008-0141-y](https://doi.org/10.1007/s10044-008-0141-y)
- Gershenson JK, Prasad GJ, Zhang Y (2003) Product modularity: definitions and benefits. *J Eng Des* 14(3):295–313
- Giffin M, de Weck O, Bounova G, Keller R, Eckert C, Clarkson PJ (2009) Change propagation analysis in complex technical systems. *J Mech Des* 131(8):081001
- Gingerich PD (2001) Rates of evolution on the time scale of the evolutionary process. *Genetica* 112–113:127–144
- Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans Syst Sci Cybern* 4(2):100–107
- Henderson RM, Clark KB (1990) Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. *Adm Sci Q* 35(1):9–30
- Hirtz J, Stone R, McAdams D, Szykman S, Wood K (2002) A functional basis for engineering design: reconciling and evolving previous efforts. *Res Eng Des* 13(2):65–82
- Hölttä-Otto K, Chiriac NA, Lysy D, Suh ES (2012) Comparative analysis of coupling modularity metrics. *J Eng Des* 23(10–11):790–806
- Hubka, V (1982) General Model of Methodical Procedure During Design—The General Procedural Modell. In: Eder, WE (ed) *Principles of Engineering Design*. Butterworth Scientific, London, pp 44–76
- Hu J, Cardin MA (2015) Generating flexibility in the design of engineering systems to enable better sustainability and lifecycle performance. *Res Eng Des* 26(2):121–143
- Humphries M, Gurney K, Prescott T (2006) The brainstem reticular formation is a small-world, not scale-free, network. *Proc R Soc B Biol Sci* 273(1585):503–511
- Jarratt T, Eckert C, Caldwell N, Clarkson P (2011) Engineering change: an overview and perspective on the literature. *Res Eng Des* 22(2):103–124. doi:[10.1007/s00163-010-0097-y](https://doi.org/10.1007/s00163-010-0097-y)
- Kong F, Ming X, Wang L, Wang X, Wang P (2009) On modular products development. *Concur Eng* 17(4):291–300
- Kotha S, Srikanth K (2013) Managing a global partnership model: lessons from the Boeing 787 ‘dreamliner’ program. *Glob Strategy J* 3(1):41–66. doi:[10.1111/j.2042-5805.2012.01050.x](https://doi.org/10.1111/j.2042-5805.2012.01050.x)
- Maier JR, Fadel GM (2006) Understanding the complexity of design. In: Braha D, Minai AA, Bar-Yam Y (eds) *Complex engineered systems: science meets technology*. Springer, Berlin, pp 122–140
- Mathieson JL, Summers JD (2010) Complexity metrics for directional node-link system representations: Theory and applications. In: *ASME international design engineering technical conferences and computers and information in engineering conference*, vol. 5: 22nd international conference on design theory and methodology; special conference on mechanical vibration and noise. ASME, pp 13–24
- Mathieson JL, Shanthakumar A, Sen C, Arlitt R, Summers JD, Stone R (2011) Complexity as a surrogate mapping between function models and market value. In: *ASME international design engineering technical conferences and computers and information in engineering conference*, vol. 9: 23rd international conference on design theory and methodology; 16th design for manufacturing and the life cycle conference. ASME, Washington, DC, pp 55–64
- McNerney J, Farmer JD, Redner S, Trancik JE (2011) Role of design complexity in technology improvement. *Proc Natl Acad Sci* 108(22):9008–9013
- Meehan J, Duffy A, Whitfield R (2007) Supporting ‘design for re-use’ with modular design. *Concurr Eng* 15(2):141–155
- Mikkola JH, Gassmann O (2003) Managing modularity of product architectures: toward an integrated theory. *IEEE Trans Eng Manag* 50(2):204–218
- Nagel J (2013) Guard cell and tropomyosin inspired chemical sensor. *Micromachines* 4(4):378–401
- Nagel RL, Midha PA, Tinsley A, Stone RB, McAdams DA, Shu LH (2008a) Exploring the use of functional models in biomimetic conceptual design. *J Mech Des* 130(12):121102
- Nagel RL, Vucovich JP, Stone RB, McAdams DA (2008b) A signal grammar to guide functional modeling of electromechanical products. *J Mech Des* 130(5):051101. doi:[10.1115/1.2885185](https://doi.org/10.1115/1.2885185)

- Nakamura T, Basili VR (2005) Metrics of software architecture changes based on structural distance. In: 11th IEEE international symposium on software metrics. IEEE Computer Society, Washington, DC, pp 24–33
- Newman MEJ (2010) Networks: an introduction. Oxford University Press, New York
- Niese ND, Singer DJ (2014) Assessing changeability under uncertain exogenous disturbance. *Res Eng Des* 25(3):241–258. doi:10.1007/s00163-014-0177-5
- Pahl G, Beitz W (1999) Engineering design: a systematic approach. Springer, Berlin
- Peterson M, Mocko GM, Summers JD (2012) Evaluating and comparing functional and geometric complexity of products. In: ASME international design engineering technical conferences and computers and information in engineering conference, vol 7: 9th international conference on design education; 24th international conference on design theory and methodology. ASME, Chicago, IL, USA, pp 551–558
- Riesen K, Emmenegger S, Bunke H (2013) A novel software toolkit for graph edit distance computation. In: Kropatsch WG, Artner NM, Haxhimusa Y, Jiang X (eds) Graph-based representations in pattern recognition. Lecture Notes in Computer Science, vol 7877. Springer, Berlin, pp 142–151. <http://www.fhnw.ch/wirtschaft/iwi/gmt>
- Sanchez R, Mahoney JT (1996) Modularity, flexibility, and knowledge management in product and organization design. *Strateg Manag J* 17:63–76
- Sarkar S, Dong A, Henderson JA, Robinson PA (2014) Spectral characterization of hierarchical modularity in product architectures. *J Mech Des* 136(1):13
- Sen C, Caldwell BW, Summers JD, Mocko GM (2010) Evaluation of the functional basis using an information theoretic approach. *Artif Intell Eng Des Anal Manuf* 24(01):87–105
- Singh V, Skiles SM, Krager JE, Wood KL, Jensen D, Sierakowski R (2009) Innovations in design through transformation: a fundamental study of transformation principles. *J Mech Des* 131(8):081010
- Sinha K, de Weck OL (2013) Structural complexity quantification for engineered complex systems and implications on system architecture and design. In: ASME international design engineering technical conferences and computers and information in engineering conference, vol 3A: 39th design automation conference. ASME, Portland, OR, USA, p V03AT03A044
- Smaling R, de Weck O (2007) Assessing risks and opportunities of technology infusion in system design. *Syst Eng* 10(1):1–25
- Sosa ME, Mihm J, Browning T (2011) Degree distribution and quality in complex engineered systems. *J Mech Des* 133(10):101008
- Stone R (2014) Design repository. Design Engineering Lab, Oregon State University, Corvallis, OR. <http://design.engr.oregonstate.edu/repo>
- Stone RB, Wood KL (2000) Development of a functional basis for design. *J Mech Des* 122(4):359–370
- Sudin M, Ahmed-Kristensen S (2011) Change in requirements during the design process. In: Culley SJ, Hicks B, McAloone T, Howard T, Dong A (eds) Proceedings of the 18th international conference on engineering design (ICED '11), vol 10: design methods and tools part 2. Design Society, Lyngby/Copenhagen, pp 200–208
- Suh NP (2001) Axiomatic design: advances and applications. Oxford University Press, New York
- Suh E, de Weck O, Chang D (2007) Flexible product platforms: framework and case study. *Res Eng Des* 18(2):67–89. doi:10.1007/s00163-007-0032-z
- Suh ES, Furst MR, Mihalyov KJ, de Weck O (2010) Technology infusion for complex systems: a framework and case study. *Syst Eng* 13(2):186–203
- Summers JD, Shah JJ (2010) Mechanical engineering design complexity metrics: size, coupling, and solvability. *J Mech Des* 132(2):021004. doi:10.1115/1.4000759
- Talay MB, Calantone RJ, Voorhees CM (2014) Coevolutionary dynamics of automotive competition: product innovation, change, and marketplace survival. *J Prod Innov Manag* 31(1):61–78
- Toensmeier PA (2005) Advanced composites soar to new heights in Boeing 787. *Plast Eng* 61(8):8
- Ulrich K (1995) The role of product architecture in the manufacturing firm. *Res Policy* 24(3):419–440
- Umeda Y, Kondoh S, Shimomura Y, Tomiyama T (2005) Development of design methodology for upgradable products based on function–behavior–state modeling. *Artif Intell Eng Des Anal Manuf* 19:161–182. doi:10.1017/S0890060405050122
- Uyeda JC et al (2011) The million-year wait for macroevolutionary bursts. *Proc Natl Acad Sci* 108(38):15908–15913. doi:10.1073/pnas.1014503108
- Uzzi B, Spiro J (2005) Collaboration and creativity: the small world problem. *Am J Sociol* 111(2):447–504
- Vincent JF, Bogatyreva OA, Bogatyrev NR, Bowyer A, Pahl AK (2006) Biomimetics: its practice and theory. *J R Soc Interface* 3(9):471–482
- Watts DJ (1999) Small worlds: the dynamics of networks between order and randomness. Princeton University Press, Princeton
- Weaver JM (2008) Innovative energy harvesting technology for wireless bridge monitoring systems. Ph.D. thesis, The University of Texas at Austin, Austin, TX, USA. <http://hdl.handle.net/2152/ETD-UT-2011-08-3956>
- Weaver JM, Wood KL, Crawford RH, Jensen D (2011) Exploring innovation potential opportunities in energy harvesting using functional modeling approaches. In: ASME international design engineering technical conferences and computers and information in engineering conference (IDETC/CIE2011), vol 9: 23rd international conference on design theory and methodology; 16th design for manufacturing and the life cycle conference. ASME, Washington, DC, pp 479–489
- Wyatt DF, Eckert CM, Clarkson PJ (2009) Design of product architectures in incrementally developed complex products. In: Norell Bergendahl M, Grimheden M, Leifer L, Skogstad P, Lindemann U (eds) DS 58-4: proceedings of the 17th international conference on engineering design (ICED 09), vol 4. The Design Society, Stanford, CA, USA, pp 167–178
- Wyatt DF, Wynn DC, Clarkson PJ (2013) A scheme for numerical representation of graph structures in engineering design. *J Mech Des* 136(1):011010. doi:10.1115/1.4025961