A Network Approach to Define Modularity of Components in Complex Products

Manuel E. Sosa[1]
Technology and Operations Management Area,
INSEAD,
77305 Fontainebleau, France
e-mail: manuel.sosa@insead.edu

Steven D. Eppinger
Sloan School of Management,
Massachusetts Institute of Technology,
Cambridge, Massachusetts 02139

Craig M. Rowles
Pratt & Whitney Aircraft,
East Hartford, Connecticut 06108

Modularity has been defined at the product and system levels. However, little effort has gone into defining and quantifying modularity at the component level. We consider complex products as a network of components that share technical interfaces (or connections) in order to function as a whole and define component modularity based on the lack of connectivity among them. Building upon previous work in graph theory and social network analysis, we define three measures of component modularity based on the notion of centrality. Our measures consider how components share direct interfaces with adjacent components, how design interfaces may propagate to nonadjacent components in the product, and how components may act as bridges among other components through their interfaces. We calculate and interpret all three measures of component modularity by studying the product architecture of a large commercial aircraft engine. We illustrate the use of these measures to test the impact of modularity on component redesign. Our results show that the relationship between component modularity and component redesign depends on the type of interfaces connecting product components. We also discuss directions for future work. [DOI: 10.1115/1.2771182]

1 Introduction

Previous research on product architecture has defined modularity at the product and system levels [1–3]. However, little effort has been dedicated to studying modularity at the component level [4]. Although complex products are typically considered as networks of components that share interfaces to function as wholes [5–7], no quantitative measures distinguish components based on how connected (or disconnected) they are with other components in the product. The term "modularity" has been used to imply decoupling of building blocks, such that the more decoupled the building blocks of a product or system, the more modular that product or system is [1,8]. We provide an alternative notion of modularity at the component level by examining components' design interface patterns with those of other components within the product rather than their internal configuration. More specifically, we define measures to quantify the relative level of modularity of components in complex products based on their lack of connectivity with other components within the product.

Understanding architectural properties, such as component modularity, is particularly important for established firms, which often fail to identify and manage novel ways in which components may share interfaces [9]. Managing interfaces becomes even more difficult when developing complex products; hence, it is critical for managers to proactively identify those components that will require particular attention during the design process [10,11]. Many important design decisions depend on how the components connect with other components in the product, yet we still lack accepted measures to capture how disconnected (i.e., how modular) a component is. Do modular components require more (or less) attention from their design teams during their development process? Are modular components easier to redesign or outsource? In order to answer such questions, we propose to quantitatively measure modularity at the component level.

The need to measure modularity has been highlighted implicitly by Saleh [12] in his recent invitation "to contribute to the growing field of flexibility in system design" (p. 850). Saleh [12] laments that "there isn't yet a coherent set of results that demonstrates how to embed flexibility in the design of complex engineering systems, nor *how to evaluate it* and trade it against other system attributes such as performance or cost" (p. 849, emphasis added). Defining and measuring modularity at the component level (as opposed to the product or system level) represents an important step in addressing this void in the engineering design literature because it can provide quantitative approaches to *evaluate* the flexibility associated with components embedded in complex products. Our proposed definitions of component modularity therefore may serve as starting points for a long-needed discussion about architectural properties of product components.[2]

Our work makes two important contributions. First, we integrate the literature of product architecture, social networks, and graph theory to define and measure modularity at the component level based on the notion of centrality. We apply our definitions to determine the modularity of the components of a large commercial aircraft engine. Second, we illustrate how to test the impact of component modularity on important design decisions, such as component redesign. In particular, we show that the relationship between component modularity and component redesign is not trivial and depends on the type of design interface that connects the product components. Our approach illustrates how to study the relationship between component modularity and other important performance or life cycle attributes of product components.

2 Literature Review

This work builds upon streams of research in both product architecture and social networks. We also refer to graph theory, which provides a foundation for defining properties of both products and social networks when they are considered as graphs of connected nodes. We blend these research streams together by defining and measuring three types of component modularity.

---

[1]Corresponding author.

[2]We refer to architectural properties of product components as those determined by the components' patterns of interfaces with other components in the product.

Copyright © 2007 by ASME

2.1 Product Architecture. The literature on product decomposition and product architecture begins with Alexander [13], who described the design process as involving the decomposition of designs into minimally coupled groups. Simon [5] elaborated further by suggesting that complex systems should be designed as hierarchical structures consisting of "nearly decomposable systems," such that strong interfaces occur within systems and weak interfaces occur across systems. This is consistent with the independence axiom of axiomatic design, which suggests the decoupling of functional and physical elements of a product [6]. Taking a more strategic view, Baldwin and Clark [8] argued that modularity adds value by creating options that enable the evolution of both designs and industries.

Ulrich [1] defined the architecture of a product as the "scheme by which the function of a product is allocated to physical components" (p. 419). A key feature of the product architecture is the extent to which it is modular or integral [14]. In the engineering design field, significant research has focused on rules to map functional models to physical components [15,16] and decomposition methods of complex products using graphs, trees, and matrices [17–19]. This line of research has paid particular attention to the identification of clusters of similarly dependent components (also called *modules*). As for measures of modularity, most previous work concentrates on the product level [4], such that existing modularity measures consider similarity and dependency links between product components [19–22].

As Ulrich [1] suggested, establishing the product architecture involves not only the arrangement of functional elements and their mapping onto physical components but also the specification of the interfaces among interdependent components. In order to capture the structure of product architectures in terms of component dependencies, we use the design structure matrix (DSM) tool, a matrix-based graphical method introduced by Steward [23] and used by Eppinger et al. [24] to study the interdependence between product development activities. Furthermore, DSM representation has been used to document product decomposition and team interdependence [3,11,25,26] and to model the risk of design change propagation in complex development efforts [10,27–29]. More recently, researchers have extended the use of DSM representations of complex products to analyze their architectures at the product level [30,31].

2.2 Social Networks. A social network refers to a set of actors connected by a set of ties. The actors, or "nodes," can be people, groups, teams, or organizations, and the ties are social relationships such as friendship, advice, or communication frequency. Social network analysis studies the social relations among a set of actors and argues that the way an individual actor behaves depends in large part on how that actor is tied into the larger web of social connections [32,33]. This research also postulates that the success or failure of societies and organizations depends on the interactions of their internal entities [34]. Beginning in the 1930s, a systematic approach to theory and research began to emerge when Moreno introduced the ideas and tools of sociometry [35]. In the 1940s, Bavelas [36] noted that the arrangement of ties linking team members may have consequences for their productivity and morale, and proposed that the relevant structural feature to study was *centrality*. Since then, social network analysis has extended into many different areas of organizational research [37].

The work most relevant to our paper is that which focuses on developing network measures to capture structural properties of social systems at the individual and group levels. Of particular relevance is work focused on developing centrality measures of individual actors in social organizations [33,38,39]. Actors who are the most important (also referred to as prominent or prestigious actors) are usually located in "central" locations within the network. Thus, centrality measures aim to identify "the most important" actors in a social network based on their social interactions [33,38]. Although many measures of node centrality have
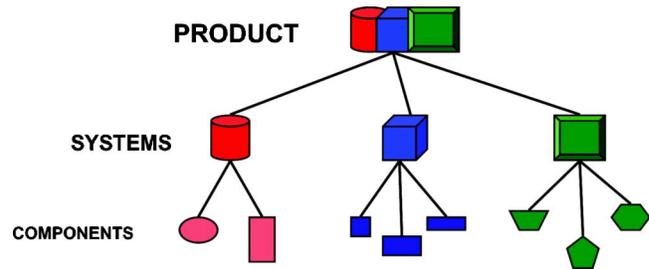

Fig. 1 Hierarchical decomposition of a product

been suggested, it was not until Freeman's [38] article that clarity about the concept and general ways to measure it converged into three categories of centrality: degree, closeness, and betweenness. We discuss these three categories in detail when we develop our component modularity measures.

In addition to centrality, other measures of social network properties—such as power, constraint, range, and redundancy—exist, but their translation to the product domain is less obvious [33,34]. Algorithms to compute most of these structural properties are available and have been implemented in network computer programs such as UCINET [40].

2.3 Graph Theory. Graph theory [41,42] has been used widely in social network analysis [33,38,39,43] and, to a lesser extent, in engineering design [17,18,44,45]. The most salient benefits of using graph theory to study networks include, first, a common language to label and represent network properties and, second, mathematical notions and operations with which many of these properties can be quantified and measured [33] (p. 93).

Before developing measures of component modularity, we must first clarify some basic graph theoretical concepts [41,42]. A *graph* is a collection of points (also called vertexes or nodes) and lines (also called arcs, ties, linkages, or edges). In our context, the components of a product are represented by the nodes of a graph, and the "connections" among these components are represented by the edges of the graph. The *degree* of a node is the number of edges incident with it. A *path* is a sequence of distinct, connected nodes in a graph, and the *length* of a path is the number of edges on it. In turn, a *geodesic* is the shortest path between two nodes, and the geodesic distance, or simply the *distance*, between two nodes is the length of their geodesic. A graph is *connected* if every pair of nodes is joined by a path. A *bridge* is an edge whose removal would disconnect the original graph into separate subgraphs. The *center* of a connected graph is the node (or set of nodes) with the smallest maximum distance to all other nodes in the graph [42] (p. 46). A *star* graph consists of one node at the center and some number of nodes, each of which is connected to the center node and to no other node [33]. Finally, when the edges of the graph have arrows, allowing for asymmetric as well as symmetric relations between nodes, the graph is *directed* (also called a *digraph*), and the preceding definitions may be extended easily to take the directionality of the edges into account.

3 Defining Component Modularity

The term "modularity" has received widespread attention across various disciplines [1–3,8,21,46,47], but, thus far, confusion remains about its definition and ways to measure it [2]. In order to measure modularity, we must clarify the various levels of analysis on which the term can be defined, which is particularly relevant when designing complex products due to their decomposition into systems and components [5]. In Fig. 1, we show how a product can be decomposed into several systems, which can be decomposed further into components. Modularity, therefore, can be defined at the product, system, and component levels.

At the product level, Ulrich [1] defined modular product architecture as resulting from a one-to-one mapping between functional
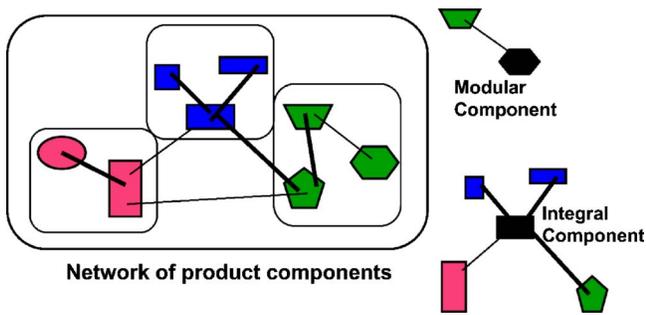
Fig. 2 Network representation of a product

elements and physical components and including "de-coupled component interfaces" (p. 423). At the system (or subsystem) level, much work has focused on clustering similarly dependent components together that are tightly connected inside the cluster and loosely connected with other clusters [4,5,16,18,22,25]. Moreover, Sosa et al. [3] defined modular systems "as those whose design interfaces with other systems are clustered among a few physically adjacent systems" (p. 240). Herein, we define and measure modularity at the component level.

Therefore, to define component modularity, we analyze each component's network, as defined by its connections with all other components in the product. Formally, we define component modularity as the *level of independence of a component from the other components within a product*. The more independent (or disconnected) a component is (i.e., the more "degrees of freedom" a component has), the more modular it is. We assume that components lose design independence because of their connections with other components, which we call *design dependencies*.[3] As a result, we aim to measure component modularity by considering the patterns of a component's design dependencies with the other components in the product.

Figure 2 shows a network view of a hypothetical product decomposition, in which we have added component dependencies to the hierarchical structure in Fig. 1. Figure 2 also shows the network of the most modular and least modular components in such a network based on their lack of connectivity with the other components in the product. However, we still need a way to quantify the level of connectivity of a component within a product.[4]

In general terms, we operationalize component modularity as the ratio of actual component "disconnectivity" to the maximum disconnectivity a component could have in a product of n components. Hence,

$$\text{Component modularity} = \frac{\text{Actual component disconnectivity}}{\text{Maximum possible component disconnectivity}} \quad (1)$$

This expression offers a normalized measure of component modularity that depends on how we measure the connectivity of a component within the product. Because component modularity depends on the architecture of the product, a normalized measure is required to be able to compare the design independence of components across products. We do this based on the notion of centrality because it is one of the most widely used concepts employed in empirical studies that involve the identification of the

most important nodes of a network [33]. Freeman [38] suggested measuring centrality based on three unique properties shown by the center node of a star graph: maximum number of direct connections to all other nodes in the graph, minimum distance to all other nodes in the graph, and maximum occurrence on the path of two other nodes in the graph [33]. That is, the central node of a star graph is *directly connected* to all other peripheral components, is the *closest* node to all other nodes in the graph, and is the only node that is *between* any two other nodes in the graph.

We assume that more central (or more connected) components exhibit higher levels of some or all of these three distinct properties and, therefore, measure connectivity among product components by considering either direct, indirect, or bridging connections among them. We do this because components are not only directly connected to other components (degree connectivity) but also indirectly connected to others because design dependencies can potentially propagate through intermediary components and reach other distant components (distance connectivity), or they can also serve as bridges by connecting two other components (bridge connectivity).

3.1 Design Dependencies. In order to define modularity measures for product components formally, we first capture the breakdown structure of the product into functional or physical components, then identify the design dependencies (including types and strength) between these components, and finally model the product as a network of components to measure their level of modularity.

Previous work in engineering design identifies design dependencies between functional components on the basis of flows of energy, material, and information among functional elements of products during their concept development [6,7,48]. Other researchers identify various types of design dependencies between physical components to capture how the functionality of one physical component depends on spatial, structural, material, energy, and information constraints of other components in the product [3,25,30]. Still others capture design dependencies between components based on their *impact* on other components as a result of a likely *change* in the design of a component [10,27,29]. In addition to distinct types of design dependencies, researchers have used various discrete scales to document the strength of connections, which either enhance or reduce the functionality or performance of the component, for each dependency type [3,25,29,30].

A subtle but important issue regarding the identification and documentation of design dependencies requires determining how to deal with dependencies that may influence the product-level performance (also called system-level performance), such as the aerodynamic performance of an aircraft engine. In order to address this issue for each component design dependency identified, we suggest two alternate approaches that incorporate product-level impacts into the definition of the dependency.

- First, we would treat product-level requirements as potential external constraints on all the components of the product and ensure that such constraints are manifest in the definition of the design dependencies of the components affected by those constraints. For example, the clearance between two engine components would be defined as a strong bidirectional spatial design dependency between them if it affects the rotor dynamic performance of the engine. Note that using this approach depends on the definition of the types of design dependencies that could connect any two components [27,29]. If necessary, one could define a design dependency type that exclusively captures product-level requirements, such as weight or fuel economy, and therefore connect components exclusively in terms of product-level requirements. This would be appropriate if the requirements cannot be defined within a reasonable interpretation of standard design dependency types, such as spatial, structural, material, energy, and information [3,25,48].

---

[3]We use the expression "design dependency" to refer to a specific type of connection between two components, such as the ones defined due to spatial or energy constraints, whereas we use the expression "design interfaces" to refer to component connections in a broader sense because they are typically formed by the aggregation of design dependencies of various types.

[4]Note that we use the term *connectivity* as a property of the components of a product, whereas graph theory uses the term as an attribute of the entire graph. In graph theory, the connectivity of a graph is the minimum number of points whose removal results in a disconnected graph [41] (p. 43).

- Second, we would embed product-level requirements within "virtual" physical elements of the product and treat these as any other physical product components. This approach would enable us to capture the impact of design dependencies that propagate to nonadjacent components through such virtual components, and their contribution to our measures of component modularity would be taken into account, just as the contribution of any other component in the product would be. For example, the aerodynamic performance of an aircraft engine is integrally associated with the secondary airflow that circulates through it, and component design dependencies throughout the engine relate to the careful management of secondary airflow. Therefore, design dependencies could be defined between the engine's actual physical components and the secondary air, which instantiates to a large extent the performance requirements of the engine. In our case study, we use this second approach to validate the implementation of the first approach.

As mentioned above, the product breakdown into components and the design dependencies between them define the network of components to analyze. This network can be represented by a design dependency matrix, $\mathbf{X}$. In order to keep our nomenclature clear for the rest of this section, let $\mathbf{X}$ refer to the matrix of design dependencies for any type of design dependency, which captures the direct dependencies between components for any given design domain. (Note that $\mathbf{X}$ is simply a component-based DSM associated with a dependency type.) To be consistent with Sosa et al. [3], we maintain that $\mathbf{X}$ has nonzero elements, x_{ij}, if component i depends for its functionality on component j. The value of x_{ij} indicates the strength of the design dependency, ranging from 0 to x_{max}, and diagonal elements, x_{ii}, are defined as zero.

3.2 Degree Modularity. Our simplest definition of component modularity is *degree modularity $M(D)$*, which relates negatively to the number of other components with which a given component has direct design dependencies. The larger the number of components that affect or are affected by the design of component i is, the less modular component i is.

Because the degree of a node "is the number of lines incident with" it [41] (p. 14), it ranges from a minimum of 0 to a maximum of $(n-1)$ if there are n nodes in a graph. Since design dependencies have both direction and strength, we need to extend the concept of node degree to valued directed graphs to define degree modularity.

The *in-degree* of a component i is equal to the number of other components that i depends on for functionality, whereas *out-degree* is equal to the number of other components that depend on component i. Thus, we define, for a product with n components, the in-degree modularity of component i, $M(ID)_i$, as

$$M(ID)_i = \frac{\text{Actual indegree disconnect.}}{\text{Max. indegree disconnect.}}$$

$$= \frac{\text{Max. indegree disconnect.} - \text{Actual indegree connect.}}{\text{Max. indegree disconnect.}}$$

(2)

Hence,

$$M(ID)_i = \frac{x_{max} \cdot (n-1) - x_{i+}}{x_{max} \cdot (n-1)} = 1 - \frac{x_{i+}}{x_{max} \cdot (n-1)} \quad (3)$$

where $x_{i+} = \sum_{j=1, j\neq i}^{n} x_{ij}$ and x_{max} is the maximum value that x_{ij} can take.

Similarly, the out-degree modularity of component i, $M(OD)_i$, can be defined as

$$M(OD)_i = 1 - \frac{x_{+i}}{x_{max} \cdot (n-1)} \quad (4)$$

where $x_{+1} = \sum_{j=1, j\neq i}^{n} x_{ji}$.

The maximum degree modularity occurs when a component is not connected to any other component in the product. Moreover, $M(ID)_i$ and $M(OD)_i$ range over $[0,1]$. The minimum value of degree modularity corresponds to a component that has *strong* design dependencies with all other $(n-1)$ components of the product. Hence, such a component would be highly integral. The value of degree modularity increases linearly as the direct connectivity of a component decreases. If there are no design dependencies (either $x_{i+} = 0$ or $x_{+i} = 0$), the component is completely disconnected from others for that design dependency direction, and the resulting in- or out-degree modularity is equal to 1.

3.3 Distance Modularity. Although degree modularity captures how many other components are directly linked to component i, it does not consider any indirect ties by which component i may have design dependencies with other components in the product network. We argue that the modularity of component i also depends on how *distant* it is from all other components in the product. Closeness centrality, from the social network theory, is the concept we build upon. The closeness centrality of an actor reflects how close an actor is to other actors in the network; as Freeman [38] (p. 224) suggested, "the independence of a point is determined by its closeness to all other points in the graph." These ideas were originally discussed by Bavelas [36], but it was not until Sabidussi [43] proposed that actor closeness should be measured as a function of geodesic distance that a simple and natural measure of closeness emerged. We incorporate these ideas into the product architecture domain by using the notion of distance between components, such that the more distant a component is from the other components, the further its design dependencies have to propagate and, thus, the more modular the component is.

Formally, we define *distance modularity $M(T)$* as proportional to the sum of the geodesics of component i with all other components in the product. Distance modularity, in its simplest form, thus depends on the direction but not on the strength of the design dependencies.

Let $d(i,j)$ denote the geodesic distance of the design dependency between component i and component j. Thus, the *in-distance modularity $M(IT)_i$* is

$$M(IT)_i = \frac{\text{Actual distance disconnectivity}}{\text{Maximum distance disconnectivity}} = \frac{\sum_{j=1, j\neq i}^{n} d(i,j)}{n(n-1)}$$

(5)

Similarly, *out-distance modularity $M(OT)_i$* becomes

$$M(OT)_i = \frac{\sum_{j=1, j\neq i}^{n} d(j,i)}{n(n-1)} \quad (6)$$

where $d(j,i)$ denotes the length of the shortest path of design dependency in the other direction, and component j depends on component i.

A high value of $M(IT)_i$ or $M(OT)_i$ means that component i is far from the others and, therefore, more modular. The denominator of our index corresponds to the maximum distance of a disconnected component, so we assume that disconnected components are n steps away from all other components in the product. Hence, disconnected components have a distance modularity of 1. The minimum value of distance modularity will be $(1/n)$, which occurs when component i is adjacent to all other components (i.e., is completely integral).

Because the expressions above do not consider the strength or propagation decay of design dependencies, we consider an alternative definition of distance modularity that we called *weighted distance modularity*. With this measure, we assign to each design
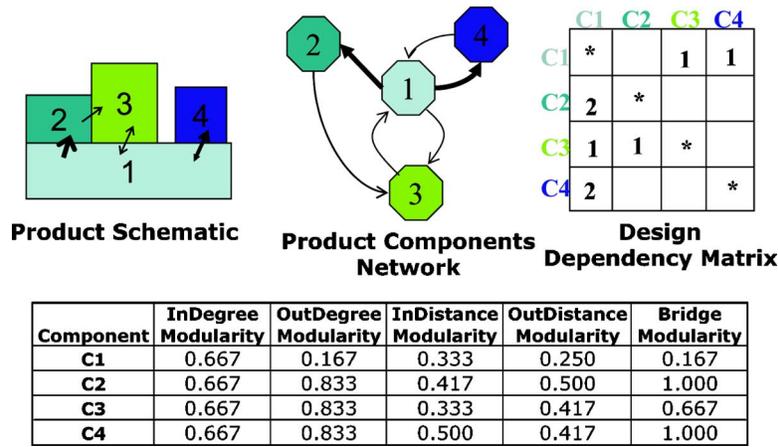
Fig. 3 Hypothetical four-component product and modularity measures of components

Component	InDegree Modularity	OutDegree Modularity	InDistance Modularity	OutDistance Modularity	Bridge Modularity
C1	0.667	0.167	0.333	0.250	0.167
C2	0.667	0.833	0.417	0.500	1.000
C3	0.667	0.833	0.333	0.417	0.667
C4	0.667	0.833	0.500	0.417	1.000

dependency a probability of propagating to other components, which is proportional to its strength. Such probabilities vary linearly from 0.0 (for design dependencies of zero strength) to 1.0 (for design dependencies of maximum strength). Then, the probability of a path between two components is equal to the product of the probabilities of the design dependencies in such a path. Finally, distance—$d_w(i,j)$—is the number of steps (i.e., number of components traversed by a design dependency) in the most probable path (*instead of the shortest path*) between components i and j. As before, we assume that disconnected components are n steps away from all other components in the product.

3.4 Bridge Modularity. A third way to measure modularity is to focus on those components that lie in the dependency path of two components. These components may control the design dependency flow because the design dependencies could propagate through them. In this sense, they can be considered bridges, or conduits that transmit design dependencies through the product network. The more a component bridges between other components, the less modular it is; that is, components may lose modularity as their bridging position increases. As a result, we define *bridge modularity* of component i based on the number of times it appears in the path between two other components.

The social network theory describes centrality in terms of the brokerage position of social actors and call it betweenness centrality. Bavelas [36] and Shaw [49] both suggested that actors located on many geodesics are central to the network, and Anthonisse [50] and Freeman [39] were the first to quantify the actor's betweenness indices.

We assume that components lying on the most geodesics are those bridging the most components and, therefore, are the least modular. This assumption makes sense in the product domain if a design dependency between two components propagates through a minimum number of parts (i.e., the geodesic). Therefore, we calculate the ratio of all geodesics between components a and b that contain component i $[nd_{ab}(i)]$ to the total number of geodesics between a and b (nd_{ab}). This comparison yields a measure of how much component i bridges between components a and b. (Note that in complex products, some components may be connected by many geodesics; therefore, an intermediary component might lie on more than one geodesic between a given pair of components.) Summing over all pairs of components a and b in the product gives us a measure of the bridging potential of component i.

Our measure of bridge modularity $M(B)$ then takes the form

$$M(B)_i = \frac{\text{Actual bridge disconnectivity}}{\text{Maximum bridge disconnectivity}}$$

$$= 1 - \frac{\sum\limits_{i \neq a, i \neq b, a \neq b} nd_{ab}(i)/nd_{ab}}{[(n-1)(n-2)]} \qquad (7)$$

The maximum bridge disconnectivity occurs when a component does not bridge any other pair of components because it is not on any of the $(n-2)(n-1)$ maximum possible paths between the other $(n-1)$ components (not including component i). In contrast, a component reaches a minimum bridge modularity of 0 only when it is at the center of a star-shaped configuration with bidirectional ties to all peripheral components [39]. The fewer geodesics are on which component i appears, the higher the value of $M(B)_i$ is and the more modular component i is.

We consider the proposed measures of component modularity complementary to each other because they emphasize related but distinct features of the patterns of design dependencies between product components. In order to illustrate this, Fig. 3 shows the product schematic and network representation of a hypothetical product with four (physical or functional) components. For simplicity, we assume that all design dependencies shown are of the same type (e.g., spatial or material) and that dependencies represented in the figure by thick edges are twice as strong as thin-edged design dependencies. Some dependencies are directional (or asymmetric) because empirical evidence shows that design dependencies may occur from one component to another, but not vice versa [3,29]. Figure 3 also shows the corresponding design dependency matrix and the modularity measures for each component.

As for degree modularity, Fig. 3 shows that because all four components have the same amount of direct inward dependencies (i.e., in-degree=2), they are equally modular from an in-degree perspective. However, component 1 is the least modular from an out-degree perspective because all other components depend on it. In general, degree modularity only takes into account the effects of immediate neighbors and neglects the connections beyond those adjacent components. Because design dependencies are not necessarily symmetric [3,29], we define in-degree and out-degree modularity. The lower the component connectivity, the more modular the component is because it is more independent of its adjacent components. Distance modularity, however, captures the effect of indirect design dependencies (due to design propagation)
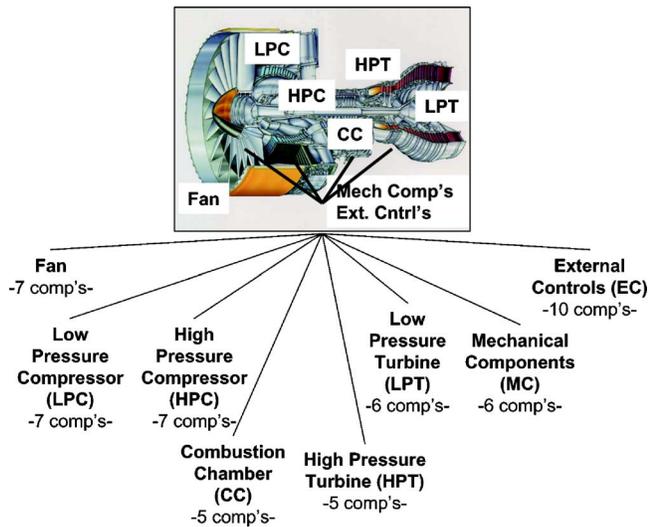
Fig. 4   PW4098 commercial aircraft engine studied

by quantifying the distance to all other components in the product. Therefore, the farther apart a component is, the more modular it is. Similar to degree modularity, we must distinguish between in-distance and out-distance modularity to take into account the direction of propagation of design dependencies. For example, Fig. 3 shows that component 4 is the most in-distance modular component because it is six steps away from being reached by all other components. (We use the term "six steps" to refer to the sum of the geodesic distance between component 4 and the other three product components. Hence, component 4 can be reached in one step by component 1, in three steps by component 2, and in two steps by component 3. To obtain our standardized distance modularity measure, we divide by 12, the maximum total distance of a component in a product with four components, which occurs only if a component is disconnected from all other components and is four steps away from each of them.) From an out-distance perspective, component 2 is the most modular because it can reach all other components in six steps, more than any other component in the product. We also determined weighted distance modularity measures assuming probabilities of 1.0 for strong design dependencies and probabilities of 0.5 for weak design dependencies, and the results are identical to the ones shown in Fig. 3 because the most probable paths coincide with the geodesics. Finally, bridge modularity is based on the component's role in bridging other components such that the fewer bridging roles a component plays, the more modular it is. This measure assumes binary design dependencies. Our example from Fig. 3 shows that both components 2 and 4 are highly bridge modular because they do not lie on the geodesic of any two other pairs of components. In contrast, component 1 lies on five out of the six possible geodesics between the other three components, which makes it the least bridge modular component.

Although defining these component modularity measures is important to advance our understanding of product architectures, some crucial questions remain to be answered: Can we assume that various design dependencies are independent of one another? What relative weight should each design dependency receive? Are modular components less likely to fail than less modular components? Are they more or less likely to be redesigned? In the next two sections, we illustrate how we address such important questions empirically.

4 Measuring Component Modularity in a Complex Product

This section illustrates how to compute and use component modularity measures in a complex product, such as a large com-

mercial aircraft engine. First, we discuss how component modularity measures correlate across various design dependencies. Then, in the next section, we discuss the link between component modularity and component redesign.

4.1 Data. We apply our network approach to analyze the modularity of the components of a large commercial aircraft engine, the Pratt & Whitney PW4098. According to our interviews with systems architects at the research site, the engine is decomposed into eight systems, each of which is further decomposed into five to ten components, for a total of 54 components. We show the hierarchical decomposition of the engine in Fig. 4. Because this was the third engine derived from the same basic system design, the product decomposition into systems and components was well understood by our informants and corresponded with the level of granularity used to establish the organizational structure that designed each of the 54 components.

After documenting the general decomposition of the product, we identify the network of design dependencies among the 54 components of the engine. We distinguish five types of design dependencies to define the design interfaces of the physical components (Table 1). In addition, we use a five-point scale to capture the level of criticality of each dependency for the overall functionality of the component in question (Table 2). Although we discuss these metrics at length in Sosa et al. [3], it is important to emphasize that this scale enables us to capture both positive and negative design dependencies. That is, our informants can identify dependencies between components that either enable or hinder the component's functionality [29]. For the purposes of our analysis, we consider three levels of criticality, indifferent (0), weak (-1, $+1$), and strong (-2, $+2$), because we assume that negative component interactions indicate equally important design dependencies to be addressed as positive ones. This assumption is consistent with our observations during the data collection. For example, we determined that the outer air seals and transition ducts (OAS-Duct) of the low-pressure turbine (LPT) impose a strong, one-directional, negative energy dependency on the LPT blades, driven by geometry and clearances between the components, which make it difficult for the blades to maintain an adequate vibration margin. On the other hand, the blades of the high-pressure turbine (HPT) have a strong, positive, bidirectional material codependency with the HPT vanes, driven by proper inlet and exit gas flow conditions

	Spatial		Structural		Material		Energy		Information	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1. In-degree	0.85	0.09	0.91	0.07	0.89	0.07	0.95	0.05	0.97	0.07
2. Out-degree	0.85	0.08	0.91	0.06	0.89	0.08	0.95	0.04	0.97	0.05
3. In-distance	0.04	0.01	0.05	0.01	0.10	0.12	0.37	0.39	0.83	0.22
4. Out-distance	0.04	0.01	0.05	0.01	0.10	0.18	0.37	0.16	0.83	0.18
5. Bridge	0.97	0.03	0.97	0.04	0.97	0.05	0.97	0.04	0.9999	0.0005

that optimize the aerodynamic efficiency of the airfoils. These design dependencies are considered equally critical for the cognizant design teams, even though the former hinders component functionality whereas the latter enables it.

Regarding the impact of engine-level requirements on design dependencies, these requirements were managed by six additional "integration" teams that were not in charge of the design of any physical engine component but, instead, were responsible for areas such as aerodynamics and rotor dynamics of the engine (see Sosa et al. [11] for details). An important responsibility of these teams was to identify and help manage design dependencies among components that could have an impact on engine performance. For example, when studying the energy dependencies between the components of the fan system, we found that the reduction of noise produced by the fan blades (a system-level requirement) drives the airfoil and platform design of both fan blades and fan exit guide vanes, resulting in a strong, bidirectional energy dependency between these components. Another example emerges from the establishment of the clearance between the tips

Fig. 5 Ego network for MC-oil pump component (spatial design dependencies)

Fig. 6 Ego network for EC-air system (spatial design dependencies)

of the HPT blades and the HPT blade outer air seals (BOAS), a symmetrical, strong, spatial dependency that must also be managed for optimum engine aerodynamic performance.

In general, engine-level requirements were cascaded down into components and, in turn, to their design dependencies of various types; therefore, we did not need to define an additional design dependency type to capture engine-level requirements exclusively. However, we note the additional challenge posed by the aerodynamic requirements of the engine. Although these requirements were also passed on to the component interface level, the secondary air team (responsible for managing all secondary airflow to optimize engine aerodynamic performance) would take the perspective of "owing the air" passing through the engine to manage some of these requirements. In this case, because the air is a physical element that passes through the engine, we consider it as a physical component of the engine and define design dependencies with it, which enables us to evaluate its additional impact on the modularity of the 54 physical engine components in terms of their connections with the secondary airflow circulating in the engine. However, we must be cautious in doing so because we risk double counting the aerodynamic requirements already captured in the design dependencies between actual engine components. In order to test the robustness of our results, we completed our analyses with secondary airflow both included and excluded from the network of components. The results we obtained after including the secondary airflow as a virtual component largely coincide with our main analysis with only the 54 physical engine components and do not change the analytical results in any significant way.
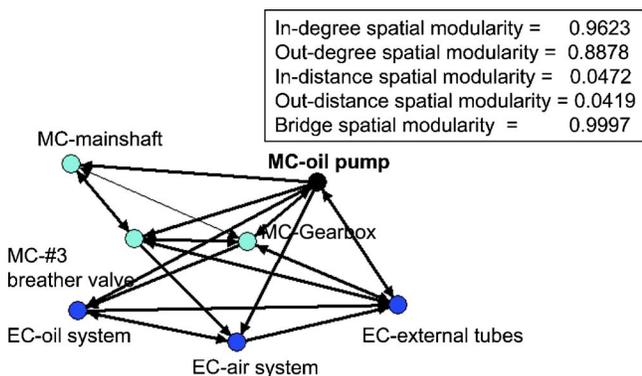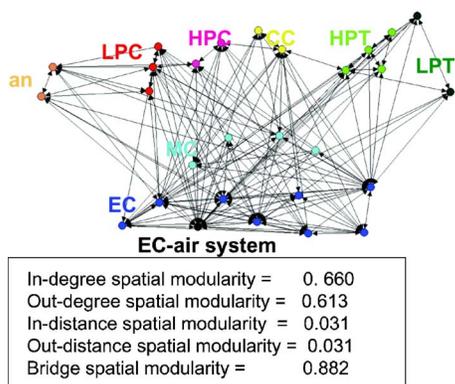
4.2 Modularity of Engine Components. In this section, we calculate and interpret the modularity measures for the engine components. Our measures follow the definitions provided previously. (As for distance modularity, we only report measures based on our original definitions, yet our results are consistent when using weighted distance modularity because of the high correlation between these two sets of measures.) Descriptive statistics are shown in Table 3. Note that distance modularity measures exhibit larger coefficients of variation both within and across design dependency types,[5] which indicates that these measures are more sensitive to small changes in product configurations than are degree and bridge modularity measures.

In order to illustrate the variation in component network configurations associated with low and high component modularity, in Figs. 5 and 6, we exhibit the "ego" network of components with low and high modularity scores for spatial design dependencies,[6] namely, the mechanical component (MC)-oil pump and the external control (EC)-air system. Nodes with the same color indicate that such components belong to the same system and arrows indicate the dependencies' directionality. In Fig. 5, the edges' thickness indicates dependency strength. In Fig. 6, we do not distinguish the strength of dependencies nor include node labels to

---

[5]The coefficient of variation of a random variable is a unitless measure of variability equal to the standard deviation divided by the mean.

[6]The ego network of component i only shows the other components it directly shares dependencies with as well as the dependencies among them.

Table 4 Partial correlation coefficients between modularity measures

	Spatial				Structural			
	1	2	3	4	1	2	3	4
1. In-degree	1.0				1.0			
2. Out-degree	0.770[a]	1.0			0.536[a]	1.0		
3. In-distance	0.812[a]	0.686[a]	1.0		0.802[a]	0.552[a]	1.0	
4. Out-distance	0.629[a]	0.832[a]	0.798[a]	1.0	0.585[a]	0.821[a]	0.721[a]	1.0
5. Bridge	0.734[a]	0.842[a]	0.624[a]	0.681[a]	0.766[a]	0.808[a]	0.658[a]	0.687[a]

	Material				Energy				Information			
	1	2	3	4	1	2	3	4	1	2	3	4
1. In-degree	1.0				1.0				1.0			
2. Out-degree	0.833[a]	1.0			0.688[a]	1.0			0.846[a]	1.0		
3. In-distance	0.247	0.211	1.0		0.519[a]	0.409[a]	1.0		0.572[a]	0.556[a]	1.0	
4. Out-distance	0.302[b]	0.309[b]	0.691[a]	1.0	0.168	0.338[b]	0.233	1.0	0.451[a]	0.702[a]	0.606[a]	1.0
5. Bridge	0.791[a]	0.833[a]	0.131	0.171	0.614[a]	0.673[a]	0.455[a]	0.155	0.858[a]	0.829[a]	0.354[a]	0.299[b]

[a]Correlation significant at the 0.01 level (two tailed).
[b]Correlation significant at the 0.05 level (two tailed).

maintain the clarity of the diagram.

When examining the direct spatial dependencies of each product component, we find that the oil pump component (which belongs to the MC system) is the most modular component from an in-degree perspective because it has only two direct (strong) spatial dependencies—with the gearbox and external tube components (see Fig. 5). From an out-degree perspective, the oil pump is less modular because there are six other components with strong spatial dependencies on it. Distance modularity scores provide additional insights about the oil pump; both in- and out-distance modularity scores have close to average values, indicating that spatial dependencies from many other nonadjacent components can reach (or be reached by) the oil pump through intermediary components (see Table 3). For the engine studied, all components are spatially connected and can reach (or be reached by) each other components through a finite number of intermediary components. In particular, the oil pump can reach all other components in 120 steps and can be reached by all other components in 135 steps, whereas the most modular component from a spatial distance point of view can reach all other components in 183 steps and can be reached by all others in 173 steps. Finally, in examining the spatial bridge modularity scores, we find that the oil pump is the fourth most modular component and, therefore, appears on very few geodesics that link any two other components. More specifically, the oil pump is only on 0.923 geodesics between any two given components. (To determine this number, we first calculate the fraction of geodesics between any two other components that contain the oil pump. Then, we sum this fraction for all pairs of components excluding the oil pump, which results in 0.923 [38,39].)

Figure 6 shows the ego network of the EC-air system component (which belongs to the externals and controls—EC—system), a highly integral component according to its many direct and indirect spatial dependencies with other components. This component is the least modular from an out-degree (spatial) perspective, as it has 22 adjacent components that spatially depend on it (19 strong dependencies) and is more modular from an in-degree perspective because it (spatially) depends directly on 20 other components (16 strong dependencies). Distance and bridge modularity measures provide similar results; the EC-air system and EC external tubes rank as the least modular components from distance and bridge perspectives for spatial dependencies.

4.3 Correlation Analysis. The preceding examples illustrate how the measures work for a particular component for a particular design dependency type. We next study how these measures relate to each other both within and across design dependency types. Therefore, we perform two correlation analyses. First, we analyze

the extent to which modularity measures differ from one another within each design dependency type (Table 4). This is important because if correlations are high between component modularity metrics for all dependency types, we might be able to use only a subset of the component modularity metrics. Second, we study the extent to which modularity measures help us highlight the differences (and similarities) between design dependency types (Table 5). This is also important because this can provide empirical evidence to justify the identification and use of all five design dependency types separately.

Table 4 shows the partial linear correlation coefficients among all measures for each design dependency. We find significantly positive correlation coefficients among all measures of component modularity for spatial, structural, and information design dependencies. That is, within spatial, structural, and information design dependency domains, our modularity measures greatly coincide in their assessments of component modularity. Correlation coefficients are less significant for material and energy design dependencies, particularly with respect to several of the distance modularity measures. For example, within the material domain, the variation of in-distance modularity is not strongly associated with the variation of (in- or out-) degree modularity nor with that of bridge modularity. Similarly, within the energy domain, the variation of out-distance modularity is not strongly associated with the variation of in-degree modularity or of bridge modularity. Because distance modularity captures how components are connected not only with neighboring components but also with all other components in the product, this result suggests that material and energy design change propagations would follow paths that are not strongly associated with direct dependencies, which, in turn, are better captured by degree and bridge modularity measures. Before discussing the implications of these results for the engine we studied, let us consider the second correlation analysis.

Table 5 shows the partial correlation coefficients among the five design dependencies for all measures of component modularity. In general, the results show a significantly strong correlation between spatial and structural component modularity (for all measures of modularity), whereas material, energy, and information dependencies evince weaker and/or less significant correlation coefficients, particularly for distance and bridge modularity measures. This finding provides important empirical evidence that the modularity of a component should not be based on only one type of design dependency.

Additional empirical evidence from our study is consistent with the results of these correlation analyses. In our case study, many materials and energy design dependencies did not necessarily correspond with other types of design dependencies. For example,

Table 5 Partial correlation coefficients between design dependencies

	In-degree				Out-degree			
	1	2	3	4	1	2	3	4
1. Spatial	1.0				1.0			
2. Structural	0.751[a]	1.0			0.674[a]	1.0		
3. Material	0.527[a]	0.218	1.0		0.779[a]	0.443[a]	1.0	
4. Energy	0.617[a]	0.564[a]	0.208	1.0	0.565[a]	0.392[a]	0.315[b]	1.0
5. Information	0.620[a]	0.415[a]	0.194	0.711[a]	0.570[a]	0.080	0.359[a]	0.604[a]

	In-distance				Out-distance				Bridge			
	1	2	3	4	1	2	3	4	1	2	3	4
1. Spatial	1.0				1.0				1.0			
2. Structural	0.836[a]	1.0			0.844[a]	1.0			0.741[a]	1.0		
3. Material	0.274[b]	0.114	1.0		−0.011	−0.102	1.0		0.733[a]	0.402[a]	1.0	
4. Energy	0.188	0.131	0.205	1.0	0.422[a]	0.469[a]	−0.090	1.0	0.537[a]	0.431[a]	0.559[a]	1.0
5. Information	0.501[a]	0.359[a]	0.134	0.183	0.669[a]	0.487[a]	0.021	0.189	0.132	−0.044	0.035	0.249

[a]Correlation significant at the 0.01 level (two tailed).
[b]Correlation significant at the 0.05 level (two tailed).

the design of many MC of the oil system depend on many other components for the material transfer of oil. However, their design is less dependent on other components for spatial, structural, and energy requirements. In addition, material and energy dependencies may be more subjective and difficult to identify than structural and spatial dependencies. For example, blade design depends on pressure profiles of gases flowing from the vanes (material dependency), and these are less likely to be considered as design dependencies than the required clearance between them (spatial dependency). In other cases, many design dependencies are unidirectional; for example, blade designs for vibration margin (an undesired energy dependency) depend on the number of upstream vanes, but not vice versa. These empirical observations are consistent with the observed lack of significant correlations across measures for energy and material dependencies (Table 4) and across dependency types for distance modularity measures (Table 5).

5 The Relation Between Component Modularity and Component Redesign

In the previous section, we performed a descriptive analysis of the three proposed measures of component modularity. Yet, what can these measures be used for? In addition to using these measures to rank components according to their level of disconnectivity within the product, we can also use them to enhance our understanding of performance-related attributes of product components. We can do this by theorizing and testing the relationships between component modularity and component design decisions. This is important for managers and engineers when making decisions about product components that depend on their connectivity with other components within the product. Some of these decisions include component engineering outsourcing, mitigation of component obsolescence, and component redesign [10,12]. We use our modularity measures in this section to build a new understanding of how component modularity impacts component redesign decisions. In this context, we define *component redesign* as the percentage of actual novel design content relative to the design of the component included in the previous version of the product.

Previous work in engineering design has studied design changes in complex products [10,27,29,51]. Yet, the link between modularity and redesign is not well understood [52]. We formulate two important but conflicting hypotheses that link component modularity and component redesign based on the assumption that design changes propagate across components as a result of their connectivity through various types of design dependencies [53]. Similar to previous work in engineering design [27,52], we dis-

tinguish between initiated and emergent design changes, which result in planned and unplanned redesigns, respectively.

An important implication of design change propagation is *unplanned redesign* or design rework [54]. Because components are connected through various types of design dependencies, design changes in one component are likely to propagate to other components in the product. As a result, a component that depends (directly or indirectly) on many other components or one in the midst of many other components is more likely to be in need of redesign to accommodate unforeseen design changes (or greater design changes than those planned) in (or required by) other components [10,53,54]. That is, the more inward interfaces a component has, the higher is the likelihood that unforeseen changes in other components will carry over onto it [52]. Hence, we formulate our hypothesis related to unplanned redesign as follows.

H1: Components with low in-degree, in-distance, and bridge modularity levels are more likely to exhibit higher levels of (unplanned) redesign due to impact from changes in other components.

A second implication of design change propagation pertains to the allocation of design changes among various components in a product. In complex products, managers and engineers need to choose which components to redesign to fulfill the functional requirements of the new product and/or adapt to planned changes in adjacent components. While doing so, engineers probably redesign those components that are less likely to impact the others. That is, components with fewer outward design dependencies on other components in the product are better candidates to be redesigned. This is consistent with the argument of Baldwin and Clark [8], who suggested that modularity fosters innovation because it decouples design teams to work on independent modules. Hence, we formulate our hypothesis regarding planned redesign as follows.

H2: Components with high out-degree and out-distance modularity levels are more likely to exhibit higher levels of (planned) redesign.

In order to test our hypotheses with our data, we needed to capture the levels of planned and unplanned redesigns of each of the 54 engine components. We were able to capture only the former by asking design teams to "provide an estimate of the level of redesign required for your parts or system for the PW4098, as a percentage of the prior existing engine design." Although we did not explicitly ask for it, we believe the answers to our question mostly capture planned redesign rather than unplanned redesign (i.e., design effort that adapts the component to a new product) because the engineers' estimates of the percentage of redesign were normalized by a common reference point (previous engine

model) and their knowledge of what it takes—and, in this case, what actually took—to adapt the parts to the new configuration (foreseen and planned changes). In the derivative engine studied, very little unplanned redesign of major significance occurred or was required. During follow-up interviews to validate our data, we identified two important sources of unplanned redesign that happened during development after the initial detail designs were released to make the first development parts. Still, the estimates of the percentage of redesign of the components involved did not change because of the nature of the rework; the parts were redesigned already and had to be done over (i.e., the amount of work performed was much higher, but not much more percentage of redesign of these components occurred).

Because our component redesign data only capture planned redesign, we can test only our second hypothesis (*H2*). To do so, we estimate the multivariate nonlinear model specified below. Our dependent variable is a fraction; so, estimating an ordinary least squares (OLS) linear model may be problematic because the predicted values from an OLS regression are never guaranteed to fall within the unit interval, which can result in biased coefficient estimates. In addition, the coefficient of a linear model assumes that the effect of a predictor variable is constant across all levels of the dependent variable, which again may not be accurate. There are several ways to address these issues. A common solution is to estimate a linear model for the log-odd ratio of the dependent variable, though this involves adjusting observations on extreme values [55] (p. 402). A better alternative, proposed by Papke and Wooldridge [56], does not require any data adjustment. We estimate our models with such a procedure in STATA-SE 9 using generalized linear models (GLMs) with family (binomial), link (logit), and robust standard errors. Note that we estimate the model adjusting standard errors for intragroup correlation using the cluster procedure implemented in STATA to take into account that components were architected into eight systems, which suggests that observations within a given system may not be independent. To test the robustness of our results, we estimated linear and semilog functional forms and obtained analogous results to the ones presented herein,

$$E(\text{component redesign of component } i | \mathbf{x})$$

$$= G[\beta_0 + \beta_{\text{spatial}}(\text{spatial modularity of component } i)$$

$$+ \beta_{\text{structural}}(\text{structural modularity of component } i)$$

$$+ \beta_{\text{material}}(\text{material modularity of component } i)$$

$$+ \beta_{\text{energy}}(\text{energy modularity of component } i)$$

$$+ \beta_{\text{info}}(\text{information modularity of component } i)] \quad (8)$$

Component redesign is the dependent variable of interest whose variation we want to explain with component modularity measures for all five types of design dependencies. $G[\cdot]$ is the logistic function and β's are the partial effects that indicate the strength of the impact of each type of component modularity on the dependent variable (see Papke and Wooldridge [56] for details). Because there are five proposed metrics of component modularity, each emphasizing a distinct aspect of it, we estimate our model for each measure and thereby test whether the proposed modularity measures for each design dependency have a significant relationship with component redesign. The results of our regression analysis are shown in Table 6, with GLM estimates for each model. To assess the goodness of fit of our models, we follow the recommendation provided by Papke and Wooldridge [56] (p. 629) and estimate the models' coefficient of multiple determination,[7] R^2.

----

[7]$R^2 = 1 - \text{SSE}/\text{SST} = 1 - \Sigma(y_i - \hat{y}_i)^2 / \Sigma(y_i - \bar{y}_i)^2$, where y_i is the observed dependent variable, $\bar{y}_i$ is the mean of the dependent variable over the 54 observations, and $\hat{y}_i$ is the predicted value of the dependent variable obtained from our regression models. This index estimates the proportion of the total variation in the dependent variable that is explained by the regression model.

Table 6 Effects of component modularity on component redesign (*N*=54).

	Model 1, in-degree	Model 2, out-degree	Model 3, in-distance	Model 4, out-distance	Model 5, bridge
Constant	−4.706 (3.913)	1.052 (5.646)	−2.185[a] (1.198)	−1.780 (1.892)	−356.453[a] (195.315)
Spatial	−3.193 (3.446)	5.743 (4.744)	47.912 (66.241)	**244.405**[b] **(31.919)**	−6.538 (10.354)
Structural	5.281[c] (2.338)	−4.811 (3.740)	−3.194 (53.412)	**−156.312**[b] **(30.137)**	6.077 (5.364)
Material	3.687 (3.091)	3.078 (2.940)	0.137 (0.736)	−0.100 (1.045)	6.046 (4.022)
Energy	−6.076 (7.501)	−6.612 (8.178)	0.030 (.492)	1.585 (1.431)	−4.727 (4.509)
Information	5.170 (4.302)	1.941 (6.384)	0.252 (1.245)	−1.958 (1.573)	355.676[a] (198.342)
R^2	0.064	0.116	0.062	0.250	0.038

[a]$p < 0.1$.
[b]$p < 0.01$.
[c]$p < 0.05$.
Robust standard errors adjusted for the eight clusters in the system appear in parentheses. These values are used to determine confidence intervals around the regression coefficients. For example, the 95% confidence interval for the estimate of the coefficient for out-distance spatial modularity (Model 4) ranges from 181.846 to 306.945.

Models 1–5 estimate component redesign using various component modularity types. Not surprisingly, Models 1, 3, and 5 exhibit the least explanatory power (i.e., lowest R^2), consistent with the empirical observation that our dependent variable captures planned redesign. Model 2 shows higher explanatory power, but Model 4 offers the best goodness of fit to our data (i.e., highest R^2). As a result, we concentrate our discussion on Model 4.

Model 4 includes significant coefficients for spatial and structural dependencies. The significantly positive spatial coefficient indicates that the more modular components (from an out-distance perspective) in the spatial domain are the more likely to exhibit higher levels of redesign. That is, components that are less likely to transmit spatial dependency to others (because they are more distant to other components) are more likely to exhibit higher levels of redesign, consistent with our second hypothesis (*H2*). Interestingly, Model 4 also shows that structural out-distance modularity negatively impacts component redesign, which appears (at least at first) not to support *H2* because it indicates that components that are more likely to transmit forces and loads to other components (i.e., less modular from a structural out-distance viewpoint) are more likely to exhibit higher levels of redesign. Finding such opposite effects for component redesign when measuring modularity based on the same criteria (out-distance modularity) is an apparent paradox.

However, the results do not conflict if we distinguish between *desired* and *undesired* design propagation. Generally, spatial dependencies refer to linkages that can disrupt the design of other components if they propagate through. As a result, we expect engineers to avoid redesigning components that are tightly integrated with other components. This is consistent with Baldwin and Clark's [8] view of modularity, which emphasizes decoupling of components (i.e., modularization) to avoid disruption and encourage innovation within modules. However, there is an alternative view of the effects of modularity and innovation that relates to performance maximization [1,31]. This view postulates that integrality is necessary to better fulfill functional requirements. That is, to meet the new functional requirements of the engine, some dependencies are more likely to be propagated intentionally across components. According to our results, these desired dependency propagations are more likely to correspond to structural dependencies in our case study. In order to understand these results, we must put them in the context of the development of this engine.

The PW4098 was a derivative engine, which, by definition, required redesigning only those systems and components necessary to achieve the new, higher level of performance. The main functional requirement driving engine performance was the increase of engine thrust, which entailed the intentional transmission of greater longitudinal forces through the engine. This was achieved, in short, by increasing fan and turbine capacity, thus running the high-pressure core faster and hotter. As a result, components related to bearings, fuel, oil, and airflow (with structural, material, and energy transfers) had increased performance specifications and were redesigned as required. On the other hand, redesign of some components with stronger spatial dependencies was avoided because they tend to be more disruptive and largely pertain to "competition" for common space. These results support the view that designers are more likely to concentrate design changes on components that are more distant from a spatial viewpoint yet structurally closer to many other components. For example, the fan (a system with, on average, more than 70% redesign) is structurally linked to all the cases and rotor systems of the engine but not spatially linked to all of them. In contrast, some mechanical load components, such as bearings and shafts, that are spatially close to many other engine components but do not impact others through structural dependencies exhibited less than 10% redesign.

Another component that illustrates our results well is the HPT first blade (25% component redesign), which has more spatial than structural constraints and with those spatial constraints being very "expensive" to change. The blade airfoil length is set by the engine flow path, as it is defined going through that stage in the HPT. Changing the flow path would likely cascade into changes required in virtually every part in the HPT, as well as potentially the rest of the engine flow path. This proposal, therefore, is far more complex and extensive than forcing the blade airfoil length to remain unchanged and dealing with the related disadvantages of that decision. In this case, the increased speed and temperature of the engine core increased loads on the blade, rotor, and case structure. The engineers, in turn, responded with improved cooling configurations and reinforced structures, as appropriate. The axial and radial clearance changes (gapping) were also minimized for similar reasons.

These results illustrate the importance of using various component modularity measures to capture different aspects related to the connectivity of components in a complex product. In our case study, only out-distance modularity is meaningful in studying how engineers allocate redesign decisions. Therefore, we finish our discussion with some comments about the definition of our modularity measures. First, note that our three measures of component modularity linearly depend on centrality measures. An important advantage of using a linear functional form to describe the relationship between modularity and centrality is that nonlinear functions can be specified when regression models are estimated using component modularity as predictor variables. That is, if researchers think that a certain component attribute depends in a nonlinear fashion on component modularity, they can still use our measures and stipulate such nonlinearity in their regression model formulation.

6 Conclusions and Future Work

This paper enhances our understanding of product architecture concepts by providing formal definitions and measures of modularity at the component level. We take a network approach to define three measures of component modularity based on centrality measures originally developed to study social networks [38]. Our definitions of component modularity emphasize various aspects of modularity relevant at the component level. Degree modularity is negatively proportional to the number and strength of design dependencies with adjacent components; distance modularity is proportional to the mean distance with all other components in the product; and bridge modularity is negatively

proportional to the number of bridging positions that a component occupies in the dependency network. We quantify and interpret these measures for all five types of design dependencies documented for the components of a large commercial aircraft engine. We also illustrate how to use component modularity measures to empirically understand component performance metrics, such as component redesign.

Using our component modularity measures we test whether redesign efforts are concentrated on more modular components. In our case study analysis, we find that modular components are favored for allocating design changes that can disrupt the design of other components, whereas integrally connected components are favored for design changes associated with fulfilling key functional requirements. Although we cannot claim the generality of these results without completing similar studies with other types of products in different industries, we expect to obtain analogous findings that explain the link between component modularity and component redesign in other complex products, such as computers, automobiles, and airplanes.

Our work also highlights the importance of considering both dependency structure and design change as integral aspects of the process to develop complex products [24,57]. We also illustrate the challenges associated with doing so. We note some limitations to our way of modeling design dependencies, one of which is that we only address product specifications that are manifest in the design dependencies. Another important limitation is that the network of design dependencies is likely to be incomplete because we rely on design experts to reveal them for us. In our comparison with the communication network formed by the teams that designed the 54 components of the engine studied, we find substantial mismatches between design interfaces and team interactions, possibly because many design dependencies were not known in advance (see Sosa et al. [11] for details). Computer-based engineering tools may help future research in this area by offering tools that facilitate the documentation of design dependencies in complex products.

Having quantitative ways to determine the architectural position of a component within the product is particularly relevant in complex products comprised of many components that share interfaces along various design domains. Establishing the relationship between component modularity and product performance metrics (beyond component redesign explored herein) remains an interesting challenge for future work. Are modular components less likely to fail than integral components? Which type of component modularity is a better predictor of component failure? Because component modularity is based on its connectivity within a product, the same component can have different modularity measures across products. How does component modularity affect component sourcing and quality?

In this paper, we study component modularity for a single product and do not explore how component modularity changes over time. Quantitative approaches that can capture component modularity easily will be useful to track these measures across several product generations. Doing so can enhance our understanding of how changes in the architecture of the product affect the network properties of each component.

Although we believe that our three proposed measures of component modularity have substantial meaning and are relatively simple to calculate (once the network of component design interfaces has been documented), we also recognize that future efforts should develop alternative measures that capture other architectural properties of components based on how they share design interfaces. How can we combine these measures to attain an aggregated measure of component modularity? How can we extend these concepts to the system and product levels? How do architectural properties, such as component modularity, relate to the social network properties of the organizations that develop them? Our ongoing research efforts focus on answering some of these questions [58].

Finally, this work opens new opportunities for research in the area of engineering design by combining product architecture representations and social network analysis. We have benefited from previous studies of centrality measures of social networks. Other social network concepts that may also merit further research by the engineering design community include structural equivalence, group cohesion, structural holes, and social influence. How can we adapt these concepts to develop better product architectures?

Acknowledgment

References

[1] Ulrich, K. T., 1995, "The Role of Product Architecture in the Manufacturing Firm," Res. Policy, **24**, pp. 419–440.
[2] Gershenson, J. K., Prasad, G. J., and Zhang, Y., 2003, "Product Modularity: Definitions and Benefits," J. Eng. Design, **14**(3), pp. 295–313.
[3] Sosa, M. E., Eppinger, S. D., and Rowles, C. M., 2003, "Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions," ASME J. Mech. Des., **125**(2), pp. 240–252.
[4] Gershenson, J. K., Prasad, G. J., and Zhang, Y., 2004, "Product Modularity: Measures and Methods," J. Eng. Design, **15**(1), pp. 33–51.
[5] Simon, H. A., 1981, The Science of the Artificial, 2nd ed., MIT, Cambridge, MA.
[6] Suh, N. P., 2001, Axiomatic Design: Advances and Application, Oxford University Press, New York.
[7] Kusiak, A., 1999, Engineering Design: Products, Processes, and Systems, Academic, New York.
[8] Baldwin, C. Y., and Clark, K. B., 2000, Design Rules: Volume 1: The Power of Modularity, MIT, Cambridge, MA.
[9] Henderson, R., and Clark, K., 1990, "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," Adm. Sci. Q., **35**(1), pp. 9–30.
[10] Clarkson, P. J., Simons, C. S., and Eckert, C. M., 2004, "Predicting Change Propagation in Complex Design," ASME J. Mech. Des., **126**(5), pp. 765–797.
[11] Sosa, M. E., Eppinger, S. D., and Rowles, C. M., 2004, "The Misalignment of Product Architecture and Organizational Structure in Complex Product Development," Manage. Sci., **50**(12), pp. 1674–1689.
[12] Saleh, J. H., 2005, "Perspectives in Design: The Deacon's Masterpiece and the Hundred-Year Aircraft, Spacecraft, and Other Complex Engineering Systems," ASME J. Mech. Des., **127**(3), pp. 845–850.
[13] Alexander, C., 1964, Notes on the Synthesis of Form, Harvard University Press, Cambridge, MA.
[14] Ulrich, K., and Eppinger, S. D., 2004, Product Design and Development, 3rd ed., McGraw-Hill, New York.
[15] Kirschman, C. F., and Fadel, G. M., 1998, "Classifying Functions for Mechanical Design," ASME J. Mech. Des., **120**, pp. 475–482.
[16] Stone, R. B., Wood, K. L., and Crawford, R. H., 2000, "A Heuristic Method for Identifying Modules for Product Architectures," Des. Stud., **21**(1), pp. 5–31.
[17] Michelena, N., and Papalambros, P. Y., 1995, "A Network Reliability Approach to Optimal Decomposition of Design Problems," ASME J. Mech. Des., **117**(3), pp. 433–440.
[18] Chen, L., and Li, S., 2005, "Analysis of Decomposability and Complexity for Design Problems in the Context of Decomposition," ASME J. Mech. Des., **127**, pp. 545–557.
[19] Wang, B., and Antonsson, E. K., 2005, "Hierarchical Modularity: Decomposition of Function Structures With Minimal Description Length Principle," Proceedings of IDETC/CIE 2005, 17th Design Theory and Methodology Conference, Paper No. DETC2005-85173.
[20] Gershenson, J. K., Prasad, G. J., and Allamneni, S., 1999, "Modular Product Design: A Life-Cycle View," J. Integr. Des. Process Sci., **3**(4), pp. 13–26.
[21] Newcomb, P. J., Bras, B., and Rosen, D. W., 1998, "Implications of Modularity on Product Design for the Life Cycle," ASME J. Mech. Des., **120**, pp. 483–490.
[22] Lai, X., and Gershenson, J. K., 2006, "Representation of Similarity and Dependency for Assembly Modularity," Proceedings of the 2006 ASME Design Engineering Technical Conferences, 18th International Conference on Design Theory and Methodology, Philadelphia, PA.
[23] Steward, D., 1981, "The Design Structure Matrix: A Method for Managing the Design of Complex Systems," IEEE Trans. Eng. Manage., **EM-28**(3), pp. 71–74.
[24] Eppinger, S. D., Whitney, D. E., Smith, R. P., and Gebala, D. A., 1994, "A Model-Based Method for Organizing Tasks in Product Development," Res. Eng. Des., **6**(1), pp. 1–13.
[25] Pimmler, T. U., and Eppinger, S. D., 1994, "Integration Analysis of Product Decompositions," ASME Conference on Design Theory and Methodology, Minneapolis, MN, pp. 343–351.
[26] Browning, T. R., 2001, "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions," IEEE Trans. Eng. Manage., **48**(3), pp. 292–306.
[27] Eckert, C. M., Clarkson, P. J., and Zanker, W., 2004, "Change and Customization in Complex Engineering Domains," Res. Eng. Des., **15**(1), pp. 1–21.
[28] Browning, T. R., and Eppinger, S. D., 2002, "Modeling the Impacts of Process Architecture and Cost and Schedule Risk in Product Development," IEEE Trans. Eng. Manage., **49**(4), pp. 428–442.
[29] Jarratt, T., Eckert, C., and Clarkson, P. J., 2004, "Development of a Product Model to Support Engineering Change Management," Proceedings of the TMCE 2004, Lausanne, Switzerland, Vol. 1, pp. 331–342.
[30] Sharman, D., and Yassine, A., 2004, "Characterizing Complex Products Architectures," Syst. Eng., **7**(1), pp. 35–60.
[31] Hölttä, K., Suh, E. S., and de Weck, O., 2005, "Tradeoff Between Modularity and Performance for Engineering Systems and Products," International Conference on Engineering Design (ICED), Melbourne.
[32] Freeman, L. C., 2004, The Development of Social Network Analysis: A Study in the Sociology of Science, BookSurge, North Charleston, SC.
[33] Wasserman, S., and Faust, K., 1994, Social Network Analysis, Cambridge University Press, New York.
[34] Burt, R., 1992, Structural Holes: The Social Structure of Competition, Harvard University Press, Cambridge, MA.
[35] Moreno, J., 1934, Who Shall Survive?: Foundations of Sociometry, Group Psychotherapy, and Sociodrama, Nervous and Mental Disease, Washington, D.C.
[36] Bavelas, A., 1948, "A Mathematical Model for Small Group Structures," Hum. Organ, **7**, pp. 16–30.
[37] Borgatti, S., and Foster, P. C., 2003, "The Network Paradigm in Organizational Research: A Review and Typology," J. Manage., **29**, pp. 991–1013.
[38] Freeman, L., 1979, "Centrality in Social Networks: Conceptual Clarification," Soc. Networks, **1**, pp. 215–239.
[39] Freeman, L., 1977, "A Set of Measures of Centrality Based on Betweenness," Sociometry, **40**, pp. 35–41.
[40] Borgatti, S. P., Everett, M. G., and Freeman, L. C., 2002, UCINET for Windows: Software for Social Network Analysis, Harvard: Analytic Technologies, Cambridge, MA.
[41] Harary, F., 1969, Graph Theory, Addison-Wesley, Reading, MA.
[42] Wilson, R. J., 1996, Introduction to Graph Theory, 4th ed., Pearson Education, Essex, England.
[43] Sabidussi, G., 1966, "The Centrality Index of a Graph," Psychometrika, **31**, pp. 581–603.
[44] Park, H., and Cutkosky, M. R., 1999, "Framework for Modeling Dependencies in Collaborative Engineering Design Processes," Res. Eng. Des., **11**, pp. 84–102.
[45] Gebala, D. A., and Eppinger, S. D., 1991, "Methods for Analyzing Design Procedures," Proceedings of the ASME Third International Conference on Design Theory and Methodology.
[46] Sanchez, R., and Mahoney, J. T., 1996, "Modularity, Flexibility, and Knowledge Management in Product and Organization Design," Strategic Manage. J., **17**, pp. 63–76.
[47] Schilling, M. A., 2000, "Toward a General Modular Systems Theory and Its Application to Interfirm Product Modularity," Acad. Manage. Rev., **25**(2), pp. 312–334.
[48] Pahl, G., and Beitz, W., 1991, Engineering Design: A Systematic Approach, K. Wallace, ed., Springer-Verlag, New York.
[49] Shaw, M. E., 1954, "Group Structure and the Behavior of Individuals in Small Groups," J. Psychol., **38**, pp. 139–149.
[50] Anthonisse, J. M., 1971, The Rush in a Graph, Mathematische Centrum, Amsterdam.
[51] Fricke, E., Gebhard, B., Negele, H., and Igenbergs, E., 2000, "Coping With Changes: Causes, Findings, and Strategies," Syst. Eng., **3**(4), pp. 169–179.
[52] Eckert, C., Clarkson, J., and Earl, C., 2005, "Predictability of Change in Engineering: A Complexity View," Proceedings of IDETC/CIE 2005, 17th Design Theory and Methodology Conference, Paper No. DETC2005-85435.
[53] Mihm, J., Loch, C., and Huchzermeier, A., 2003, "Problem-Solving Oscillations in Complex Engineering Projects," Manage. Sci., **46**(6), pp. 733–750.
[54] Smith, R. P., and Eppinger, S. D., 1997, "Identifying Controlling Features of Engineering Design Iteration," Manage. Sci., **43**, pp. 276–293.
[55] Kennedy, P., 2003, A Guide to Econometrics, 5th ed., MIT, Cambridge, MA.
[56] Papke, L. E., and Wooldridge, J. M., 1996, "Econometrics Methods for Fractional Response Variables With an Application to 401(K) Plan Participation Rates," J. Appl. Econ., **11**(6), pp. 619–632.
[57] Jarratt, T., Clarkson, J., and Eckert, C., 2005, "Engineering Change," Design Process Improvement—A Review of Current Practices, Springer Verlag, London, UK, pp. 266–285.
[58] Sosa, M. E., Gargiulo, M., and Rowles, C. M., 2006, "Component Modularity, Team Network Structure, and Attendance to Technical Interdependences in Complex Product Development," INSEAD Working Paper.