Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
September 24-28, 2005. Long Beach, California, USA

DETC2005/DTM-85422

A NETWORK APPROACH TO DEFINE MODULARITY OF PRODUCT COMPONENTS

Manuel E. Sosa[1]
Assistant Professor of Technology and Operations Management
INSEAD
Fontainebleau, FRANCE

Anupam Agrawal
Technology and Operations Management
INSEAD
Fontainebleau, FRANCE

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

Craig M. Rowles
Advanced Engine Program
Pratt & Whitney Aircraft
East Hartford, Connecticut

ABSTRACT

We consider complex products as a network of components that share technical interfaces in order to function as a whole. Building upon previous work in graph theory and social network analysis, we define three measures of component modularity that consider how components may share direct interfaces with other adjacent components, how design interfaces may propagate to all other components in the product, and how components may act as "bridges" between other components. We calculate and interpret all three measures of component modularity by studying the actual product architecture of a large commercial aircraft engine. We illustrate how to use these measures to test their impact on component redesign. Directions for future work are discussed.

Keywords: Component Modularity, Product Architecture, Graph theory, Social Networks.

INTRODUCTION

Previous research on product architecture has defined modularity at the product and system level [1,2], however little effort has been dedicated to study modularity at the component level [3]. Although complex products are typically considered as a network of components that share interfaces in order to function as a whole [4,5], there are no quantitative measures that allow us to distinguish components based on how they share interfaces with other components in a product. Based on the patterns of interfaces of each component, we define measures to quantify the relative degree of modularity of components in complex products.

This paper formally defines *component modularity* based on the patterns of a component's design interfaces. 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 share interfaces [6]. Furthermore, when designing complex products it is critical for managers to proactively identify the components that will require particular attention during the design process [7]. Defining modularity at the component level (as opposed to the product level) is important because it can provide indication to managers about important component performance metrics such as design rework or failure rate. Our proposed definitions of modularity at the component level can therefore be the starting point of a long-needed discussion about architectural properties of product components.

This paper is structured as follows. First, we review the relevant literature in the product architecture, social network, and graph theory domains. Then, we define component modularity and propose specific measures. In the next section, we apply our definitions to determine the modularity of the components of a large commercial aircraft engine. Next, we define component redesign and test whether it could be predicted by component modularity. We conclude the paper with a discussion of the results and comments for future work.

LITERATURE REVIEW

This work builds upon three streams of research. The first one is the body of work dedicated to product architecture representations, and the second one is the established stream of work focused on social networks. We also build upon graph theory, which has provided the foundation to define properties of both products and social networks when considered as graphs of connected nodes. We blend these research streams

[1] Corresponding author: manuel.sosa@insead.edu
Tel: ++33 (0)1 60 72 45 36. Fax: ++33 (0)1 60 74 61 79.

together by defining and measuring three types of component modularity based on network structural properties.

Product Architecture and Graph Theory

The literature on product decomposition and product architecture begins with Alexander [8] who describes the design process as decomposition of designs into minimally coupled groups. Simon [4] elaborates further by suggesting that complex systems should be designed as hierarchical structures consisting of "nearly decomposable systems" such that strong interfaces occur within systems whereas weak interfaces occur across systems. Smith and Browne [9] describe decomposition as a fundamental approach to deal with complex engineering efforts.

Previous work has considered products as graphs of connected components. Kusiak and Wang [10] use binary digraphs representations to develop physical layouts. Moreover, component connectivity is a central concept when studying engineering changes and design propagation during the development of complex products [11,12,13,14].

Ulrich and Eppinger [15, p. 165] define the architecture of a product as "the scheme by which the functional elements of the product are arranged into physical chunks and by which the chunks interact."[2] A key feature of product architecture is the degree to which it is modular or integral. In the engineering design field, a large stream of research has focused on methods and rules to map functional models to physical components [3,16,17,18]. However, as Ulrich [1] suggested, establishing the product architecture not only involves the arrangement of functional elements and their mapping to physical components but also the specification of the interfaces among interacting components.

In order to study the structure of product architectures in terms of component interactions we use the design structure matrix (DSM) tool. The DSM is a graphical method introduced by Steward [19] and used by Eppinger et al [20] to study interdependence between product development activities. Gebala and Eppinger [21] compare DSM models with other graph based models such as program evaluation and review technique (PERT) charts and structured analysis and design technique (SADT), to study design procedures. DSM representation has also been used to document product decomposition [22] and team interdependence [23]. More recently, researchers have extended the use of DSM representations of complex products to analyze their architectures at the product level [24, 25,26] and model design change propagation [13].

Sosa et al [2] use a matrix representation not only to capture the decomposition and interfaces between product components but also to extend the concepts of product modularity to the system level. In [2] they introduced a new notion of system modularity based upon the way components share design interfaces across systems. We aim to extend this work further by defining measures that allow us to categorize components based on the direct and indirect interfaces they share with other components in the product.

[2] Ulrich and Eppinger [15] refer to chunks as the main physical building blocks which contain the physical elements of a product.

Social Networks and Graph Theory

Social network analysis is the study of social relations among a set of actors. Network analysts believe that how an individual behaves depends in large part on how that individual is tied into the larger web of social connections [27]. More importantly, they believe that the success or failure of societies and organizations often depends on the patterning of their internal structure [28]. Beginning in the 1930s, a systematic approach to theory and research, based on the above notions, began to emerge. In 1934 Jacob Moreno introduced the ideas and tools of sociometry [29]. At the end of World War II, Bavelas [30] noted that the structural arrangement of ties linking members of a task oriented group may have consequences for their productivity and morale. He proposed that the relevant structural feature was centrality, and he defined this in formal terms. Since then, social network analysis has extended into research areas that span from analysis of people in an organization to analysis of board interlocks, joint ventures and inter-firm alliances and trade blocks - drawing upon such fields as sociology, anthropology, and mathematics [27, 28, 31].

A social network is a set of actors who are connected by a set of ties. The actors or "nodes" can be people, groups, teams, or organizations. Ties connect pairs of actors and can be directed (for example, A gives advice to B) or undirected (for example, A and B are friends) and can be binary (for example, whether A gives advice to B or not) or valued (for example, frequency of interactions between A and B). A set of ties of a given type (such as friendship ties) constitutes a binary social relation, and each relation defines a different network (for example, the friendship network is distinct from the advice network).

We identify two streams of research in the field of social networks. First, and most relevant to our paper, is the work focused on developing network indices to capture structural properties of social networks at the individual level. Secondly, is the stream of work that focuses on how social network properties of individuals or teams impact the performance of organizational processes such as product development [E.g. 28, 32, 33].

Graph theory has been widely used in social network analysis [34, 35, 27, 31]. One of the main notions that social network analysis derives from graph theory is to identify the most important actors in a network. Actors who are the most important (prominence and prestigious are also commonly referred terms) are usually located in "central" locations within the network. Centrality measures aim to identify "the most important (or prominent)" actors in a social network. In our context, this would translate to identifying the most central (or most integral) components in a complex product. It is important to note that measures of centrality not only consider direct ties (between adjacent actors) but also indirect ties through intermediary actors.

Freeman [36] discusses three different measures of centrality: degree, closeness, and, betweenness. Degree centrality refers to the simplest definition of actor centrality which indicates that central actors must be the ones that have the most ties to other actors in the network, or the ones which other actors depend upon the most. A second perspective of centrality is based on how close an actor is to all the other actors in the network, implying that an actor is more central if it can quickly reach all others. Hakimi [37] and Sabidussi [38] quantify this view of centrality by suggesting that central actors have "minimum steps" when linking to all other actors. A third view on centrality is related to the role of the broker (or gatekeeper) between other actors in a social network. That is, interactions between two non-adjacent actors may depend on the other actors in the network, in particular the ones that lie in the path between the two [39]. Indices for all three centrality measures have been developed for both non-directed (symmetric) and directed (asymmetric) relations between actors.

In addition to centrality, there are other measures, such as constraint and redundancy, that quantify important individual properties based on their social network [28]. Although these properties are important to study social networks they are less relevant in the product domain. Algorithms to compute most of these structural measures are available and implemented in network computer programs such as UCINET [40].

DEFINING COMPONENT MODULARITY

The term 'Modularity' has received widespread attention across various disciplines [1, 3, 41, 42, 43]. When designing complex products, modularity is considered an important product characteristic that results from directly mapping their functional and physical components [1,15]. Moreover, Ulrich [1] defines modular product architectures as the ones resulting from a one to one mapping between functional and physical components. By considering complex products as collection of systems which are further decomposed into components, Sosa et al [2] define modularity at the system level. They define modular systems "as those whose design interfaces with other systems are clustered among a few physically adjacent systems, whereas integrative systems are those whose design interfaces span all or most of the systems that comprise the product due to their physically distributed or functionally integrative nature throughout the product". We extend this line of research by defining modularity at the component level.

In general terms, we define component modularity as the *level of independence of a component from the other components in a product*. Contrary to the definitions of modularity at product and system level, our definition at the component level implies a range of modularity. That is, the more independent a component is (i.e. the more "degrees of freedom" a component has), the more modular it is. We assume that components lose "independence" during the design process due to their interactions, which we call *design dependencies*, with other components. As a result, we aim to measure component modularity by considering the patterns of design dependencies of a component. This argument is similar to the underlying proposition in social network studies which define various structural characteristics of nodes in a social network based on their patterns of interactions.

Based on graph theory, social network research has quantified structural properties for individuals, teams, and organizations in a social network. Centrality is one of the most important structural properties in social network analysis and our starting point to define modularity at the component level. In the context of product architecture, centrality does not directly relate to modularity of a component – however there is an inverse relation. The less central a component is, the less prominent it will be – therefore there will be less design dependencies on (and from) other components. Thus, the least central component in a complex product architecture network will be a candidate for maximum modularity position. Based on this, we develop three indices for component modularity: degree, distance, and bridge modularity.

Design Dependency Matrix, X

In order to formally define modularity measures for product components we define the design dependency matrix, **X**. **X** is a square matrix whose columns and rows are identically labeled with the components of the product. Let **X** refer to the matrix of design dependencies for any type of design dependency. Previous work in engineering design has identified various types of design dependencies between components such as spatial, structural, material, energy, and information [40, 22, 2]. Hence, **X** captures the dependency between components for any given design domain. In order to be consistent with [2], we maintain that **X** has non-zero elements, X_{ij}, if component i depends for functionality on component j. The value of X_{ij} indicates the strength of the design dependency.

Degree Modularity

The simplest definition of component modularity (M) can be in terms of the number of other components with which it has direct design dependencies. Hence, we can measure modularity as the answer to the questions: "how many components' designs do we need to obtain technical information about to finalize the design of this component?", and "how many other components depend on the design of this component?" The larger the number of components that affect, or are affected by, the design of component i, the less modular component i is. Based on this rationale, the modularity of a component would be inversely proportional to the number of direct design dependencies.

As per graph theory conventions [34, 35], the degree of a node is the number of lines that are incident with it. The degree of node therefore ranges from a minimum of 0 to a maximum of (n-1) if there are n nodes in a graph. In the product architecture domain, a node is a component and an arc (i.e. link between two nodes) is a design dependency. Since design dependencies have both direction and strength we need to extend the concept of node degree to valued directed graphs in order 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

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

where $x_{i+} = \sum_{j=1}^{n} X_{ij}$ and x_{max} is the maximum value that X_{ij} can take. Note that for binary relations our measure of degree modularity is the inverse of degree centrality proposed in previous social network studies [36].

Similarly, the *Out-Degree Modularity* of component *i*, $M(OD)_i$, can be defined as

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

where $x_{+i} = \sum_{j=1}^{n} X_{ji}$

A high value of $M(ID)_i$ or $M(OD)_j$ means that there are fewer and/or weaker design dependencies and therefore the component is more modular. The minimum value of degree modularity is 1, which corresponds to a component that has strong design dependencies with all other components of the product (there are *n*-1 other components). Hence, such a component would be highly integral. The maximum finite value of degree modularity, [x_{max} * (n-1)], is reached when the component is weakly linked to the rest of the product by only one other component. If there are no design dependencies (either $x_{i+} = 0$ or $x_{+i} = 0$), our definitions indicate that the component is infinitely modular for that particular design dependency. For analysis purposes, we recommend to assign a significantly larger value of degree modularity for non-connected components. For example, in our analyses we assign degree modularity for non-connected components to be 100 times greater than its maximum possible finite value.

Distance modularity

While degree modularity captures how many other components are directly linked to component *i*, it does not consider indirect ties by which component *i* can have design dependencies with another component in the product network. Here, we argue that the modularity of component *i* also depends on how "distant" it is from all other components in the product. In social network theory, closeness centrality is the concept we build upon. Closeness centrality of an actor reflects how close an actor is to other actors in the network. As Freeman [36, 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 [30]. Yet, it was not until Sabidussi [38] proposed that actor closeness should be measured as a function of geodesic distance that a simple and natural measure of closeness emerged. (In graph theory, a geodesic is the shortest path between two nodes, and geodesic distance, or simply distance, between two nodes is defined as the length of their geodesic).

We incorporate these ideas into the product architecture domain by using the notion of "distance" between components – the more distant a component is from the other components, the further its design dependencies have to propagate, hence, the more modular the component is.

Formally, we define *Distance Modularity*, *M(T)*, to be proportional to the summation of the geodesics of component *i* with all other components in the product. Similar to degree modularity, distance modularity depends on the direction of the design dependency.

Let *d(i,j)* denote the geodesic of design dependency between component *i* and component *j*. Thus, the *In-Distance Modularity*, $M(IT)_i$, is defined as

$$M(IT)_i = \frac{\sum_{j=1}^{n} d(i,j)}{n-1}$$

The denominator of our index corresponds to the minimum sum of distances, for a component that is adjacent to all other components. This determines the lower bound for our index.

Similarly, *Out-Distance Modularity*, $M(OT)_i$, is defined as follows

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

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

A high value of $M(IT)_i$ or $M(OT)_i$ means that component *i* is far away from the others and therefore is more modular. The minimum value of distance modularity will be 1, which is reached when component *i* is adjacent to all other components (i.e. the component is completely integral). The maximum finite value that this index can take (for a connected graph) is reached when component *i* is linked to the rest of the components only through one design dependency and this adjacent component is also linked to the rest of the other components by only one design dependency, and so on. This corresponds to component *i* being at the end of a linear chain of components whose summation of geodesic is equal to $n(n-1)/2$. Note that our measure is meaningful for connected graphs (i.e. components can reach all other components in a limited number of steps- that is, the distance between any two components is finite). In a non-connected graph every point is at an infinite distance from at least one other point so the numerator of our index becomes infinite, erroneously indicating that components are perfectly modular. We overcome this limitation by assuming that non-connected components (in one design domain) are *n* steps apart from all other (*n*-1) components. Hence, the maximum value of distance modularity for a non-connected component is *n*.

Bridge Modularity

A third way of measuring modularity is to focus on those components that lie in between the dependency path of two components. We can view these components as having control over design dependency flow since information about the design dependency must propagate through them. In this sense, these components can be considered as powerful gatekeepers that regulate the amount of information transmitted in the product network for some dependencies. The more a component is "in the middle" of the other components, the less modular it is.

As suggested by graph theory, "a *bridge* is a line such that the graph containing the line has fewer components than the subgraph that is obtained after the line is removed." [27, p. 114]. We argue that components lose modularity as their bridging position increases. As a result we define *bridge modularity* of component *i* based on the number of times it is on the path of two other components.

Social network theory describes centrality in terms of the brokerage position of social actors (they call it betweenness centrality). Bavelas [30] and Shaw [45] have suggested that actors located on many geodesics are central to the network. Anthonisse [46] and Freeman [39] were the first to quantify the actor's betweenness indices.

We assume that components lying on most geodesics will be the one bridging most components and therefore the least modular. This assumption makes sense in the product domain if a design dependency between two components propagates through the minimum number of parts (i.e. the shortest path or geodesic). Hence, if we calculate the ratio of all geodesics between two components, *a* and *b*, which contain component *i* ($nd_{ab}(i)$) to the number of total geodesics between *a* and *b* (nd_{ab}) we will get a measure of how "in the middle" (between *a* and *b*) component *i* is. Note that *nd* is not the geodesic distance *d* but the total number of these geodesics between *a* and *b*. Summing over all such pairs of *a* and *b* components in the product give us a measure of the bridging potential of component *i*. We standardize this measure by taking into account all pairs of components excluding component *i*. Our measure *M(B)* then takes the form

$$M(B)_i = \frac{[(n-1)(n-2)/2]}{\sum_{a<b, i\neq a, i\neq b} nd_{ab}(i)/nd_{ab}}$$

The maximum value that the denominator can take for a set of *n* connected components is (*n*-1)(*n*-2)/2, because there can be (*n*-1) components not including *i*, which can have geodesics with (*n*-2) other components. Note that the fewer geodesics component *i* is on, the higher the value of $M(B)_i$, and the more modular component *i* is. Similar to our other measures of modularity, the minimum value of this index is 1, which is reached for a perfectly integral component that is on the geodesic of all other pairs of components. Our measure of bridge modularity indicates that component *i* is infinitely modular if there are no geodesics between any other pair of components on which component *i* is on – which means that component *i* does not bridge any two other components in the product for that particular type of design dependency. To overcome this limitation, we assign a value to components with infinite bridge modularity that it is 100 times greater than the its maximum finite value for that particular design dependency.

We consider the proposed measures of component modularity to be complementary of each other because they emphasize related but distinct features about the patterns of design interfaces between product components. *Degree modularity* only takes into account the effects of immediate neighbors neglecting the connections beyond adjacent components. In addition, it is the only measure we propose that captures the strength of the design dependency. Since our design dependency matrix is not necessarily symmetric [2], we define *In-Degree* and *Out-Degree* modularity. The lower the component degree, the more modular the component is because it is more independent from its adjacent components. *Distance modularity*, on the other hand, captures the effect of indirect design dependencies by quantifying the mean distance to all other components in the product. Hence, the further apart a component is, the more modular it is. This measure, however, does not consider the effect of the design dependency strength and has limitations for product domains with non-connected components. Similar to degree modularity, we need to distinguish between *In-Distance* and *Out-Distance* modularity to take into account the direction of "propagation" of design dependencies. Finally, *bridge modularity* is based on the component's role in bridging other components. The less bridging role a component has, the more modular it is. This measure assumes binary design dependencies and is independent of their direction.

All these three measures are based on the underlying argument that as components gain degrees of freedom by sharing less design interfaces with other components they become more modular. Consequently, less modular components are components with many direct and indirect interfaces and/or occupying bridging positions in the product. Although defining these measures is important to advance our understanding of product architectures, some important questions remain to be answered. Can we assume that various design dependencies are independent of each other? What would be the relative weight given to each design dependency? (Recall that our component modularity measures are defined for each type of design dependency, that is, for spatial, structural, material, energy, and information dependency between product components.) Are modular components less likely to fail (or to be redesigned) than less modular components? In the next two sections of the paper, we illustrate how to empirically address these important questions.

AN EXAMPLE

Data

We apply our network approach to analyze the modularity of the components of a large commercial aircraft engine, the PW4098 developed by Pratt & Whitney. The engine was decomposed into eight systems (See Figure 1). Each of these systems was further decomposed into five to ten components each, for a total of 54 components. Six of the eight systems were identified as modular systems, whereas the other two systems (mechanical components system and externals and controls system) were recognized as integrative systems because of the physically distributed and functionally integrative features of their components [2].

After documenting the general decomposition of the product, we identified design interfaces between the 54 components of the engine. We distinguished five types of design dependencies to define the design interfaces between the physical components (Table 1). In addition, we used a five-point scale to capture the level of criticality of each dependency for the overall functionality of the component in question (Table 2). We discuss these metrics at length in [2]. Note that design dependencies only refer to interactions that impact the function of the component in question. We do not consider coincidental design dependencies, which could exist between spatially adjacent components.

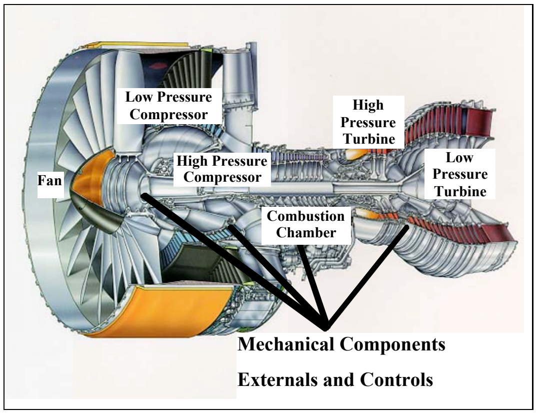

Fig 1. PW4098 Commercial Aircraft Engine Studied

Table 1. Types of design dependency

Dependency Type	Description
Spatial	Functional requirement related to physical adjacency for alignment, orientation, serviceability, assembly, or weight.
Structural	Functional requirement related to transferring loads, or containment
Material	Functional requirement related to transferring airflow, oil, fuel, or water
Energy	Functional requirement related to transferring heat, vibration, electric, or noise energy
Information	Functional requirement related to transferring signals or controls

Table 2. Level of criticality of design dependencies

Level of Criticality	Description
Required (+2)	Dependency is necessary for functionality
Desired (+1)	Dependency is beneficial, but not absolutely necessary for functionality
Indifferent (0)	Dependency does not affect functionality
Undesired (-1)	Dependency causes negative effects, but does not prevent functionality
Detrimental (-2)	Dependency must be prevented to achieve functionality

We documented our product architecture data into five design interface matrices corresponding to each type of design dependency. For the purpose of our analysis we consider three levels of criticality: Indifferent (0), Weak (-1, +1), and Strong (-2,+2). For illustration purposes, Figure 2 shows the spatial design dependency matrix of the engine studied. Figure 2 shows a square matrix (54 x 54) whose labels correspond to the 54 components comprising the engine. The non-zero cells of the matrix are marked with a "W" for weak spatial design dependencies and "S" for strong spatial design dependencies. Consistent with [2], the components labeling the rows depend (for functionality) on the components labeling the columns.

Modularity of Engine Components

In this section we calculate and interpret modularity measures for the engine components. Our measures are calculated according to the definitions provided in the previous section. For illustration purposes, we show in Table 3 the value of only one type of component modularity for each of type of design dependency.

Table 3. Some Component Modularity Measures

Component	Spatial (InDegree)	Structural (OutDegree)	Material (InDistance)	Energy (OutDistance)	Information (Bridge)
Fan Containment Case	10.6	2.79	4.87	16.8	551200
Fan Exit G. Vanes/ Cases	6.63	2.30	4.68	12.4	551200
Shroudless Fan Blades	8.15	3.79	4.75	12.3	551200
Fan Hubs	11.8	4.42	54.0	13.0	551200
Fan Stub Shafts	8.83	4.42	5.06	12.4	551200
Spinners & Nose Caps	17.7	26.5	5.30	54.0	551200
Fan Blade Platforms	10.6	13.3	5.26	13.1	551200
LPC Airfoils	6.24	5.89	4.60	18.6	551200
LPC Stator	5.30	7.57	4.57	18.5	551200
LPC Drum	13.3	6.63	5.51	19.2	551200
LPC Splitter	10.6	10.6	5.09	19.2	551200
LPC Liner	6.63	13.3	4.40	19.0	551200
2.5 Bleed (BOM)	7.57	3.12	4.30	18.3	551200
Intermediate Case	5.05	1.77	4.13	18.3	551200
HPC Blades	7.07	7.57	4.62	18.1	551200
HPC Inner Shrouds /Seals	10.6	10.6	5.49	18.4	551200
Variable Vanes	5.30	4.08	4.66	18.0	551200
HPC Fixed Stators / Cases	5.58	2.65	4.64	18.0	551200
HPC Rubstrips & Spacers	21.2	26.5	5.49	54.0	551200
HPC Disks & Drums	6.24	4.42	5.32	18.1	551200
Giggle Tube & Blds Locks	21.2	13.3	4.81	18.6	551200
Burner	8.83	4.42	4.60	18.3	551200
Diffuser	2.94	1.89	4.66	18.1	551200
Tobi Duct	8.83	6.63	5.21	18.8	551200
Diffuser Tubes	10.6	10.6	4.75	18.7	551200
Fuel Nozzle	10.6	7.57	5.17	18.3	551200
HPT Blades	5.89	26.5	4.25	16.6	551200
HPT 1V	6.63	6.63	4.77	18.9	551200
HPT 2V	8.83	10.6	4.19	16.6	551200
HPT Rotor	4.82	5.30	4.58	17.3	551200
HPT Case/OAS	3.79	3.31	4.34	18.4	551200
LP Shaft	13.25	5.89	4.55	19.6	551200
LPT Case	7.57	3.79	4.42	18.4	551200
TEC	5.05	3.79	4.51	18.4	551200
LPT Vanes	21.2	8.83	5.19	18.9	551200
LPT Blades	10.6	7.57	5.36	18.9	551200
LPT OAS / Tducts / Insl.	9.64	8.83	5.19	18.9	551200
Mainshaft IPT	3.79	3.79	4.25	18.1	551200
Gearbox	4.61	2.41	4.36	17.1	551200
#3 Breather Valve	8.15	7.57	4.75	17.8	551200
Oil Pump	26.5	6.63	4.81	17.6	551200
Intershaft Seal	13.3	26.5	5.15	54.0	551200
PMA	21.2	17.7	4.62	18.6	551200
Mech. Comp'ts/Oil System	13.3	5.89	4.23	18.1	1969
Externals Tubes	2.52	6.63	3.96	18.6	27.9
2.5 Bleed Butterfly	5.89	7.57	4.53	18.5	899
Ext./Ctrls. Air system	2.94	2.94	4.06	18.1	26.8
Ext./Ctrls. Oil system	5.05	5.89	4.38	18.6	56.1
Ext./Ctrls. Fuel/Drain	4.42	7.57	4.43	18.3	4134
Ignition	7.57	53.0	5.00	18.6	551200
Harness	2.59	4.82	4.19	18.3	36.9
Controls - Sensor	3.53	10.6	4.40	18.9	9.78
Controls - Mechanical	4.42	4.82	4.47	18.5	49.7
Controls - Electrical	4.08	13.3	4.77	18.2	5.92

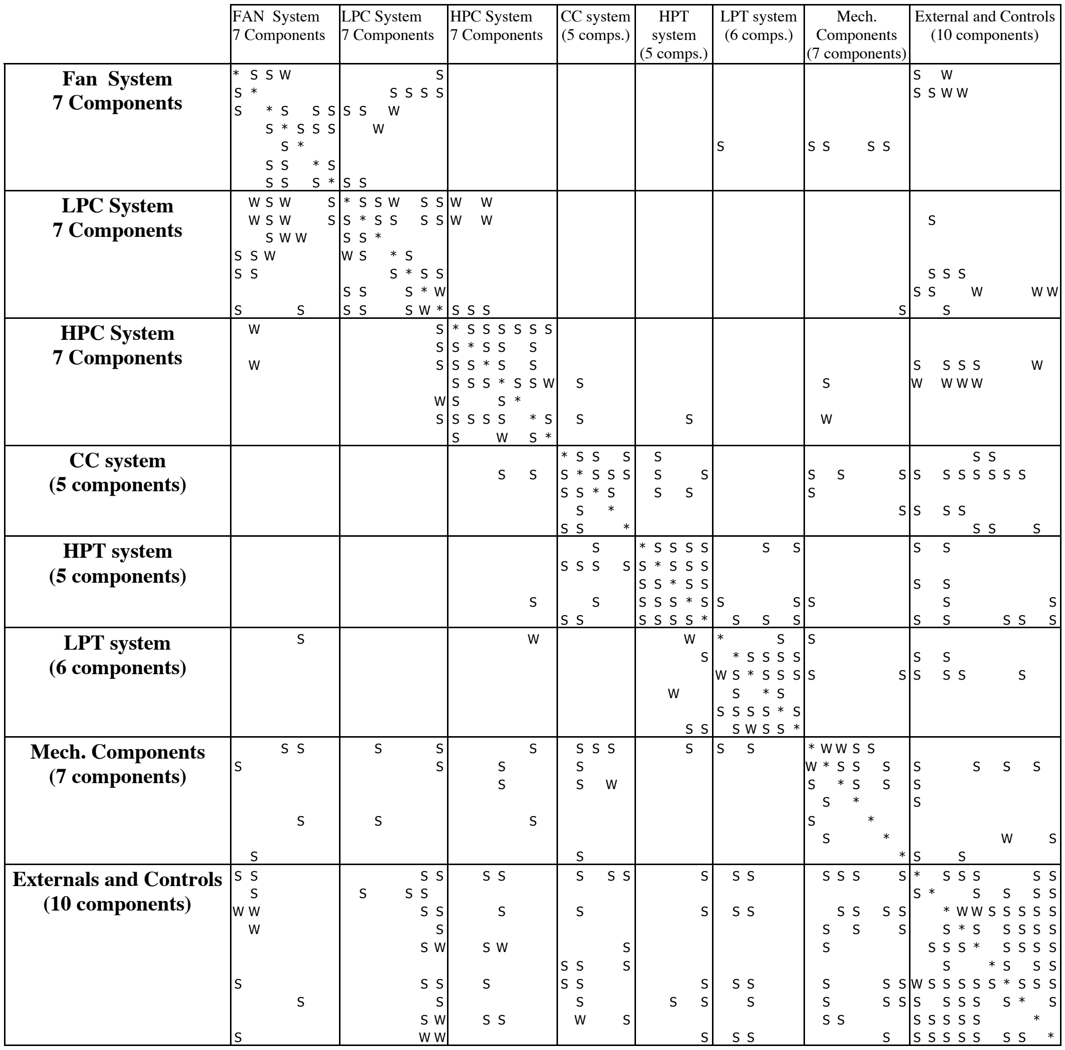

Figure 2. Spatial Design Dependency Matrix

In order to study the relation between our component modularity measures for a given design dependency as well as to understand better the relation between the various design dependencies given certain modularity measure we perform a correlation analysis.[3]

Table 4 shows the partial correlation coefficients among all the measures for each design dependency. We find significantly positive correlation among all measures of component modularity for all five types of design dependencies, except for InDistance-OutDistance energy modularity measures. To explain this last finding we visually compared the energy design dependency matrix with the other design dependency matrices and observed that energy design dependencies tend to be more unidirectional than the other types. Hence, the correlation between inflow and outflow of energy design dependencies is not significant. For example, blade designs (to meet their vibration related requirements) depend on vane passing but not the other way around.

In general, these results show that all three measures consistently indicate the level of modularity for a component. The fact that most of the design dependencies are reciprocal results in high correlation coefficients even when the measures are calculated considering the direction of design dependencies, that is, for the cases of inward and outward measures of degree and distance modularity. The strength of the correlation is relatively lower with bridge modularity, which indicates that although consistent, this measure provides a different view than degree and distance modularity.

Table 5 shows the partial correlation coefficients for all five design dependencies for all measures of component modularity. The results suggest that components have very distinct architectural properties depending on the type of dependency we examine. Given the high correlation between the modularity measures (see Table 4), we should not be surprised of observing similar correlation patterns for the various design dependencies. More importantly, we observe an overall significantly strong correlation between spatial and structural component modularity whereas material component modularity is not correlated with component modularity based

[3] Correlation analysis is concerned with measuring the strength of the relationship between variables [47].

on other types of dependencies. This provides important empirical evidence suggesting to avoid considering modularity of a component based on ONLY one type of design dependency.

In our case study, many of the materials design dependencies did not necessarily correspond to other types of design dependencies. For example, the design of many mechanical components of the oil system depended on many other components for material transferring, however their design was less dependent on other components for spatial, structural and energy requirements. Additionally, material dependencies are more difficult and subjective to assess than structural and spatial dependencies. For example, turbine blades' design depends on temperature and pressure profile of gaspath air from the turbine vanes (material dependency), which are more difficult to predict than the required clearance between them (spatial dependency). Hence, for some interfaces structural and spatial dependencies dominated the attention of system architects "neglecting" the existence of some material dependencies which were later uncovered by the design teams during the detailed design process (refer to Sosa et al [7] for details on the factors that explain the existence of these types of hidden design interfaces).

Table 4a. Partial Correlation Coefficients Between Modularity Measures

	Spatial					Structural				
	1	2	3	4	5	1	2	3	4	5
1 In-degree	1.0					1.0				
2 Out-degree	.624**	1.0				.441**	1.0			
3 In-Distance	.730**	.582**	1.0			.747**	.406**	1.0		
4 Out-Distance	.525**	.728**	.798**	1.0		.565**	.705**	.721**	1.0	
5 Bridge	.389**	.437**	.440**	.365**	1.0	.461**	.605**	.423**	.451**	1.0

** Correlation is significant at the 0.01 level (2-tailed)

Table 4b. Partial Correlation Coefficients Between Modularity Measures

	Material					Energy					Information				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1 In-degree	1.0					1.0					1.0				
2 Out-degree	.703**	1.0				.619**	1.0				.591**	1.0			
3 In-Distance	.998**	.692**	1.0			.526**	.326*	1.0			.962**	.624**	1.0		
4 Out-Distance	.701**	.997**	.691**	1.0		.463**	.790**	.233	1.0		.573**	1.00**	.606**	1.0	
5 Bridge	.434**	.615**	.444**	.617**	1.0	.446**	.372**	.537**	.378**	1	.611**	.546**	.633**	.535**	1

** Correlation is significant at the 0.01 level (2-tailed)

Table 5a. Partial Correlation Coefficients Between Design Dependencies for Each Modularity Measure

	In-degree					Out-degree				
	1	2	3	4	5	1	2	3	4	5
1 Spatial	1.0					1.0				
2 Structural	.686**	1.0				.610**	1.0			
3 Material	.077	-.056	1.0			-.081	-.098	1.0		
4 Energy	.379**	.361**	-.042	1.0		.538**	.397**	-.038	1.0	
5 Information	.301*	.063	.115	.135	1.0	.522**	.143	-.013	.188	1.0

** Correlation is significant at the 0.01 level (2-tailed)

Table 5b. Partial Correlation Coefficients Between Design Dependencies for Each Modularity Measure

	In-Distance					Out-Distance					Bridge				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1 Spatial	1.0					1.0					1.0				
2 Structural	.836**	1.0				.843**	1.0				.508**	1.0			
3 Material	.273*	.113	1.0			-.012	-.103	1.0			-.060	-.123	1.0		
4 Energy	.186	.131	.205	1.0		.420**	.469**	-.090	1.0		.369**	.589**	.138	1.0	
5 Information	.501**	.359**	.134	.183	1.0	.669**	.486**	.021	.189	1.0	.100	.196	.003	.271*	1.0

** Correlation is significant at the 0.01 level (2-tailed)

ANALYSIS: EFFECTS ON COMPONENT REDESIGN

In the previous section we performed a descriptive analysis of the three measures of component modularity we are proposing. Yet, what can we use these measures for?

Quantitative measures of component modularity could be used to predict component related performance measures. We illustrate how to do this by estimating (non-linear) models that relate component modularity to component redesign. (Our empirical models are non-linear because we must transform our bounded dependent variable in order to obtain unbiased coefficients using least squares estimation [47, p. 223].) Similar statistical models can be estimated to empirically determine the relation between component modularity measures and other performance metrics of interest such as component failure rate or engineering changes of a component.

We define component redesign as the percentage of actual novel design content relative to the previous design of such a component included in the previous version of the product. In our study, we estimated component redesign by surveying design teams about the estimated amount of redesign associated with the components they designed. The actual survey question was [48]:

"Please 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:________%."

In order to test whether component modularity can explain the variation observed in component redesign, we estimate the following multivariate regression model:

ln(*percentage redesign* of component i+1.0) = β_0 + β_1*(*spatial modularity* of component i) +
+ β_2*(*structural modularity* of component i) + β_3*(*material modularity* of component i) +
+ β_4*(*energy modularity* of component i) + β_5*(*information modularity* of component i) + ε_i

where percentage of redesign is the variable of interest whose variation we want to explain with component modularity measures for all five types of design dependencies.

The β's are the partial regression coefficients which indicate the strength of the impact of each type of component modularity on our dependent variable. For example, β_1 is interpreted as the expected change in *ln*(*percentage redesign* of component i+1.0) per unit change in spatial component modularity while the other component modularity metrics are held constant. ε's are the error terms which are assumed to be normally and independently distributed, with mean 0 and variance σ^2 [47]. Since we have five ways to determine component modularity and each of them emphasizes a distinct aspect of modularity, we estimate our regression model for all fives types of measures. We estimated the regression coefficients using the method of least squares. Note that by estimating these models we are testing whether the proposed modularity measures for each design dependency have a significant impact on component redesign.

The results of our multivariate regression analysis are shown in Table 6. Partial regression coefficients are shown for each model. We also include the adjusted R^2 for each model to indicate the proportion of the total variability in the dependent variable that is explained by the independent variables. Finally, we include the F statistic which allows us to test whether there is a significant regression relation between the response variable and the set of predictor variables [47].

Table 6. Results of Regression Analysis on Component Redesign

	Model 1 In-Degree	**Model 2 Out-Degree**	**Model 3 In-Distance**	**Model 4 Out-Distance**	**Model 5 Bridge**
Spatial	-.108***	-.103	.350	2.185**	-3.99e-6
Structural	Excluded	-.006	-.526	-1.743**	-5.6e-6
Material	Excluded	7.728e-5	.010	Excluded	3.56e-8
Energy	Excluded	.000*	.002	Excluded	-2.2-e7
Information	5.85e-5*	3.840e-5	.010	Excluded	1.54e-6
Adj R^2	0.180	-.027	-.074	.068	-.027
F	6.821***	.718	.274	2.994*	.719

*<0.1; ** < 0.05; *** <0.01

Only model 1 (based on in-degree modularity) and model 4 (based on out-distance) exhibit a significant regression relation (i.e. F statistic is statistically significant) and explain some of the variation observed in our dependent variable. Note that we excluded the variables that were not significant to increase the explanatory power of the resultant model. Doing so did not change the statistical inference of any of the coefficients shown in Table 6.

Model 1 indicates that spatial in-degree component modularity is negatively associated with component redesign. That is, the more inward spatial design dependencies a component has the higher its redesign level (relative to its previous design). This result provides empirical evidence to support the proposition that the more and stronger direct inward spatial design dependencies, the higher the level of component redesign. This proposition is sensible because one important driver of component redesign could be the number of spatially adjacent components upon which the component of interest depends. This was the case for certain systems such as the fan and turbines which were significantly redesigned to achieve the desired level of performance of this derivative engine. For example, the exit guide vanes of the fan which were over 50% redesigned depended significantly (due to spatial requirements) on several components of the low pressure compressor such as the splitter, liner, and intermediate case. On the other hand, some mechanical "supporting" components such as bearings and shafts which were less directly impacted by spatial dependencies, but instead were impacted by structural dependencies (due to transfer of loads), exhibited less than 10% redesigned.

Interestingly, model 4 (based on out-distance modularity) shows another side of the story. This model shows significant results for both spatial and structural modularity. The significantly positive spatial coefficient indicates that components with high spatial modularity are more likely to exhibit higher levels of redesign. Model 4 also shows that structural out-distance modularity negatively impacts component redesign. This 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. These results support the proposition that designers might be (intentionally or unintentionally) concentrating design changes on components that are spatially more distant and structurally closer of the rest of the components. For example, the fan (which is a system that exhibited, on average, over 70 % redesign) is structurally linked to all the cases and rotor systems of the engine but not spatially linked to all of them; the cases in turn transmit the loads to the externals and controls components some of which had to be redesigned significantly.

Interpreting our results further we assume that component redesign due to inward design dependencies (Model 1) are mostly due to architectural effects whereas component redesign due to outward design dependencies (Model 4) are mostly due to innovation effects. Architectural effects suggest that component redesign is partially driven by changes in adjacent components. Model 1 suggests that components with more spatially adjacent components are more likely to exhibit higher levels of redesign. On the other hand, innovation effects refer to the fact that some components must be inherently redesigned to achieve the desired functionality of the product. Hence, engineers must decide (to some extent) on which components to concentrate the major design changes. Model 4 suggests that engineers might choose to redesign components whose spatial dependencies do not tend to propagate beyond adjacent components. Yet, they might need to significantly redesign components that are more integral from a structural out-distance viewpoint. That is, engine components that are likely to propagate loads and containment[4] to other components are more likely to exhibit higher levels of redesign. While we cannot claim the generality of these results before completing similar studies in other types of products in different industries, we expect to obtain analogous findings to explain the link between component modularity and component redesign in other complex products such as computers, automobiles, and airplanes.

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. Our definitions of component modularity emphasize various aspects of modularity relevant at the component level. *Degree modularity* is 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. B*ridge modularity* is inversely proportional to the number bridging (or brokerage) positions that a component may have in a product 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 redesign.

Having quantitative ways to determine the architectural position of a component within the product is particularly relevant in complex products which are comprised by many components that share many interfaces along various design domains. We show how component modularity relates to component redesign. Establishing the relation between component modularity and other performance metrics remains an interesting challenge for future work. Are modular components less likely to fail than integral components? Which type of component modularity is better a predictor of component failure? Since component modularity is based on 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 have studied component modularity for one single product. We have not explored how component modularity changes over time. Having quantitative ways to easily capture component modularity will be useful to track these measures along several product generations. Doing so can enhance our understanding of how changes in the architecture of the product affects the network properties of each component.

Although we believe 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 believe that future efforts should be dedicated to develop alternative measures that capture other architectural properties of components based on how they share design interfaces. How can we combine these measures to have an aggregated measure of component modularity? How can we extend these concepts to the system and product level? How do architectural properties such as component modularity relate to social network properties of the organizations that develop them?

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

ACKNOWLEDGEMENTS

We appreciate the assistance of the engineers from Pratt & Whitney Aircraft during the development of the example presented in this paper. We also thank Martin Gargiulo for his comments on early discussions about this work.

REFERENCES

1. Ulrich, K.T. 1995. The role of product architecture in the manufacturing firm. *Res. Policy*. **24** 419-440.

[4] Propagation of certain loads is a key design requirement for engine design.

2. Sosa, M.E., S.D. Eppinger, C.M. Rowles. 2003. Identifying modular and integrative systems and their impact on design team interactions. *Journal of Mech. Design,* **125**(2) 240-252.
3. Gershenson, J.K., G.J. Prasad, and Y. Zhang. 2004. Product Modularity: Measures and Methods. *Journal of Engineering Design*, **15**(1) 33-51.
4. Simon, H.A. 1981. *The Science of the Artificial*. Second edition, MIT Press, Cambridge, MA.
5. Suh, N. P. 2001. *Axiomatic Design: Advances and Application*, Oxford University Press, New York.
6. Henderson, R., K. Clark. 1990. Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. *Admin. Sci. Quart*. **35**(1) 9-30.
7. Sosa, M.E., S.D. Eppinger, C.M. Rowles. 2004. The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. *Management Science*, **50**(12) 1674-1689.
8. Alexander, C. 1964. *Notes on the Synthesis of Form*. Harvard University Press, Cambridge, MA.
9. Smith, G. and Browne, G.1993. Conceptual Foundations of Design Problem Solving, *IEEE Transactions on Systems, Man, and Cybernetics*, vol 23, no. 5, Sept-Oct.
10. Kusiak, A. and Wang, J. 1993. Decomposition of the Design Process, *Journal of Mechanical Design*, vol. 115.
11. Fricke E., Gebhard, B., Negele, H. and Igenbergs. E. 2000. Coping with changes: causes, findings, and strategies. *Systems Engineering*, **3**(4), 169-179.
12. Clarkson, P.J., Simons, C.S., and Eckert, C.M. 2004. Predicting change propagation in complex design. *ASME Journal of Mechanical Design*, **126**(5), 765-797.
13. Jarratt, T., Eckert, C. and Clarkson, P.J. 2004. Development of a product model to support engineering change management, *TMCE 2004*, Lausanne, Switzerland, **1**, 331-342.
14. Eckert, C.M., Clarkson, P.J., and Zanker, W. 2004. Change and customization in complex engineering domains. Research in Engineering Design, **15**(1), 1-21.
15. Ulrich, K. and S.D. Eppinger. 2004. *Product Design and Development*. 3rd. Edition. McGraw Hill, New York.
16. Shapiro, V. and Voelcker, H. 1989. On the Role of Geometry in Mechanical Design, *Research in Engineering Design*, 1, pp. 69-73.
17. Kirschman, C. F. and Fadel, G. M. 1998. Classifying Functions for Mechanical Design. *Journal of Mechanical Design*. Vol 120, pp. 475-482.
18. Stone, R. B., Wood, K. L., and Crawford, R. H. 2000. A Heuristic Method for Identifying Modules for Product Architectures. *Design Studies*, Vol. 21, No. 1, pp. 5-31.
19. Steward, D. 1981. The design structure matrix: a method for managing the design of complex systems. *IEEE Trans. Eng'g. Manag.* **EM-28**(3) 71-74.
20. Eppinger, S.D., D.E. Whitney, R.P. Smith, D.A. Gebala. 1994. A model-based method for organizing tasks in product development. *Res. in Eng'g. Des.* **6**(1) 1-13.
21. 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*.
22. Pimmler, T.U., S.D. Eppinger. 1994. Integration analysis of product decompositions. *ASME Conference on Design Theory and Methodology*. Minneapolis, MN. 343-351.
23. McCord, K. R. and Eppinger, S.D. 1993. Managing the Integration Problem in Concurrent Engineering. *Sloan School of Management Working Paper Number 3594*. Massachusetts Institute of Technology.
24. Sharman and Yassine, 2004. Characterizing Complex Products Architectures. *Systems Engineering*, vol. 7, no. 1, 35-60.
25. Braha, D., Y. Bar-Yam. 2004. Topology of large-scale engineering problem solving networks," Physical Review E., **69**, 016113.
26. MacCormack, A., Rusnak, J. and Baldwin, C. 2004. Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code. *HBS Working paper 05-016*. Harvard Business School.
27. Wasserman, S., K. Faust. 1994. *Social Network Analysis*. Cambridge University Press, New York.
28. Burt, R. 1992. *Structural Holes. The Social Structure of Competition*. Harvard University Press, Cambridge, MA.
29. Moreno, J. 1934. *Who Shall Survive?: Fundations of Sociometry, Group Psychotherapy, and Sociodrama*, Washington, D.C.: Nervous and Mental Disease Publishing Co. Second edition in 1953 and Third edition in 1978.
30. Bavelas A. 1948. A mathematical model for small group structures, *Human Organization*, 7, 16-30.
31. Carrington, P. J., Scott J. and Wasserman, W. (*Eds*.) 2005**.** *Models and Methods in Social Network Analysis*. Cambridge University Press.
32. Ibarra, H. and S. B. Andrews 1993. Power, social influence, and sense making : effects of network centrality and proximity on employee perceptions. *Administrative Science Quarterly* 38: 277-303.
33. Gargiulo, M. and M. Benassi 2000. Trapped in your own net. Network cohesion, structural holes, and the adaptation of social capital. *Organization Science* 11:183-196.
34. Harary, F. 1969. *Graph Theory*. Reading, MA: Adison-Wesley.
35. Hage, P. and F. Harary. 1983. *Structural Models in Anthropology*. Cambridge: Cambridge University Press.
36. Freeman, L. 1979 . Centrality in social networks. Conceptual clarification. *Social Networks* 1:215-239.
37. Hakimi, S.L. 1965. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12, 450-459.

38. Sabidussi, G. 1966. The centrality index of a graph. *Psychometrik*a, 31, 581-603.
39. Freeman, L. 1977. A set of measures of centrality based on betweenness, *Sociometry* 40:35-41.
40. Borgatti, S.P., Everett, M.G. and Freeman, L.C. 2002. *UCINET* for Windows: Software for Social Network Analysis. Harvard: Analytic Technologies.
41. Sanchez Ron and Joseph T. Mahoney. 1996. Modularity, Flexibility, and Knowledge Management in Product and Organization Design, *Strategic Management Journal*, **17.**
42. Baldwin, C. Y., K. B. Clark. 2000. *Design rules: Volume1: The power of modularity*. Cambridge, MA: MIT Press.
43. Schilling, M.A. 2000. Toward A General Modular Systems Theory And Its Application To Interfirm Product Modularity, *Academy of Management Review*, Vol. 25, Issue 2.
44. Pahl, G. and Beitz, W., (Wallace, K., ed.). 1991. *Engineering Design, A Systematic Approach*, Springer-Verlag, New York.
45. Shaw M.E, 1954. Group Structure and the behavior of individuals in small groups, *Journal of Psychology,* 38
46. Anthonisse, J.M. 1971. *The Rush in a Graph*. Amsterdam: Mathematische Centrum.
47. Neter, J., M.H. Kutner, C.J. Nachtsheim, and W. Wasserman. 1996. *Applied Linear Statistical Models*. Fourth Edition. WCB/McGraw-Hill.
48. Rowles, C.M. 1999. *System Integration Analysis of a Large Commercial Aircraft Engine*. Master's Thesis. MIT.