Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2010
August 15 – August 18, 2010, Montreal, Quebec, Canada

IDETC 2010-28561

COMPLEXITY METRICS FOR DIRECTIONAL NODE-LINK SYSTEM REPRESENTATIONS: THEORY AND APPLICATIONS

James L. Mathieson
Research Assistant
Department of Mechanical Engineering
Clemson University
Clemson, South Carolina 29634-0921
jmathie@clemson.edu

Joshua D. Summers
Associate Professor
Department of Mechanical Engineering
Clemson University
Clemson, South Carolina 29634-0921
jsummers@clemson.edu

ABSTRACT

This paper presents an approach to defining and quantifying the complexity of systems as represented in mixed (directed and non-directed) bipartite graphs through the presentation of a central example as well as other applications. The approach presented defines nine measurements of different properties of the graph system. These measurements are derived from the representation of the system into a three dimension relational design structure matrix as well as the projections and transformations of this matrix. The metrics generated address dimensional and connective size, shortest path properties, and the decomposability of the system. Finally, a normalization and aggregate approach of these metrics is then given. This aggregation is visualized with spider graphs that facilitate viewing multiple aspects of complexity within a single perspective.

Keywords: Complexity measurement, mixed graphs, systems modeling

1 COMPLEXITY IN DESIGN

Complexity is an aspect of engineering design that is often addressed directly with the principle that "designs should be simple" [1,2]. However, such a principle fails to offer an effective means of quantifying the complexity of a given design for comparison and decision making [3]. The judgment of a design's complexity is often simply left to the individual perception of the designer. This, in turn, results in the possibility that the designs selected during the design process as being less complex by the designers may be considered more complex by the end user.

1.1. Definition of Complexity

Complexity is most often defined in the terms of systems. A system is a set of interrelated elements which, through these interrelations, manifest a behavior which the individual elements would not display independently [4]. The elements of the system can be anything capable of interaction, including molecules, consumer products, and people [5]. The foundation of complexity is the human attempt to quantify our understanding of these elements and interrelations which are counterintuitive [6].

Most definitions of complexity relate back to a measure of understanding. This leads to many views of complexity [7]. For example, a designer may be prone to defining the complexity of a product in terms of its physical components, while an end user may define complexity according to ease of use. This would suggest that the subjective definition of complexity is less of a definition and more of a perception. However, this does not mean that either of these two perceptions is incorrect. Rather, the designer here has defined the complexity by the system within the product and the user has defined complexity by the system of the product interacting with the environment. This represents the same definition of complexity, but for different system views and boundaries. Thus, complexity can be defined as the effort required to *understand* the properties of a given system [8,4].

It is for this reason that previous work has suggested the existence of multiple distinguishable factors of complexity [9,10,7]. This includes the presence of multiple attributes of complexity within a single system representation as well as multiple system representations. A system representation is defined here as the level of abstraction used in the system model and the system boundaries. The attributes of complexity take the form of various possible analytical measurements for complexity.'

This system structure approach to complexity differs from the fields of computational and information complexity. Computational complexity focuses on processes which occur within the design, requiring that connections have directionality. [11,12,13] In the opposite case, information complexity work has well established the measurement of structural complexity content within a given representation irrespective of directionality. [14,15,16]. However, the systems addressed here may have both directed connections, such as function or process models, and non-directed connections, such as physical architecture. This implies a need for metrics which capture connections in a mixed graph environment.

1.2. Matrix-Based Tools

Matrix-based system modeling techniques such as the Design Structure Matrix [17] are based on the creation of two dimensional binary matrices to represent the interrelationship that exists within a system of elements. The binary and two dimensional nature of these matrix tools does not allow for the easy distinction of relationships being represented, the possibility that two elements may be related through multiple relationships, nor that multiple elements may be related through the same relationship. The issue of relationship classification is addressed by Multiple-Domain Matrices through the use of a third "dependency type" dimension [18]. Graph-based system modeling techniques, such as those used by different engineering design representations including boundary representations, bond-graphs, and bi-partite graph based design exemplar [19], are capable of capturing all of the desired relationship information but are difficult to use in visualizing the connective properties of the system due to multiple node types [20].

An approach is developed here to measure and compare the complexity of systems in a succinct manner. First, a method for the translation of graph-based system models into a matrix-based regime is developed to facilitate complexity analysis of systems with per-instance relational details. The complexity metrics utilized capture several distinct properties of the system complexity that is present in graph-based system representations addressing both size and interconnective structure. An aggregate approach to the consideration these measures is developed. This allows for the rapid visualization of connective properties, particularly following a change in system structure.

2 EXAMPLE CASE

For the purposes of illustration, we will address an example system throughout this paper. The example system is a hypothetical group of eight designers working on a project. This hypothetical scenario is used for illustrative purposes. Thus, we will use the approaches and metrics presented in this paper to model and analyze the work interactions of this group. A system of social interactions is selected because this presents a common and easily understood situation in which a single interaction may involve multiple elements and a pair of elements may be related in more than one interaction. Other examples of graphs of interest in engineering design may be the constraint problems of the design exemplars [19], function structures [2], or component and assembly representations [10].

The interactions of the design group are modeled in terms of both meetings and workspaces. This is done to keep the example system sufficiently simple for illustration and discussion. There are three weekly meetings scheduled. Designers one, three, and five attend the first meeting; designers four, five, seven, and eight attend the second meeting; and designers two, six, and seven attend the third meeting. The design group occupies two distinct workspaces. Designers one through four work in the first workspace and designers five through eight work in the second workspace.

This can be modeled mathematically by considering the designers to form the set $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$ and the five interactions form the set $R = \{r_1, r_2, r_3, r_4, r_5\}$ where r_1 through r_5 are defined by Equations 1through 5.

Meeting 1: $$r_1 = \{e_1, e_3, e_5\} \quad (1)$$

Meeting 2: $$r_2 = \{e_4, e_5, e_7, e_8\} \quad (2)$$

Meeting 3: $$r_3 = \{e_2, e_6, e_7\} \quad (3)$$

Workspace 1: $$r_4 = \{e_1, e_2, e_3, e_4\} \quad (4)$$

Workspace 2: $$r_5 = \{e_5, e_6, e_7, e_8\} \quad (5)$$

This representation demonstrates the limitation of traditional relationship tracking tools in that the modeling of this system requires each relationship to be considered as a set of elements, rather than a singular link between elements. One approach used to handle this contradiction is to treat the relationships as nodes of a different class from those of the elements [19,21]. This is used to create a bi-partite graph representation such as that in Figure 1.

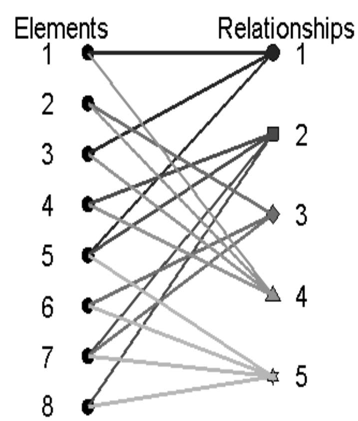

Figure 1: Bi-partite Graph

The bi-partite graph representation displays each relationship set as a series of links between the elements in the set and the relationship node. This creates a robust method of visualization for systems such as this. However, the use of two different node types and many overlapping links makes the analysis of large bi-partite graphs difficult [22]. For this reason, the system is treated as a relational design structure matrix hypergraph.

3 RELATIONAL DSM

Using sets to define each relationship necessitates the use of hypergraph representations. The relational design structure matrix (rDSM) is an array based hypergraph representation capturing relationships between multiple elements through a single instance and element pairs that are related through multiple relationship instances [9]. A third dimension is added to the traditional design structure matrix to represent the relationships as hyperedges within the hypergraph, thereby converting the hyperedge sets into design structure matrix planes along the additional dimension.

The method of construction for the hypergraph varies depending on the behavior of the interactions being modeled. If the interactions are considered to be bi-directional or without direction information, the connections modeled in each hyperedge plane will be symmetric about the diagonal as per Definition 1.

Definition 1

Given a set of elements $E = \{e_1, e_2, e_3, \ldots, e_{N_E}\}$ and a set of bi-directional relationships $R = \{r_1, r_2, r_3, \ldots, r_{N_R}\}$ where $r_i \subseteq E$, there exists a $N_E \times N_E \times N_R$ array A where $A(r_i, r_i, i) = 1$ for $i = 1{:}\,N_R$.

Here, it can be seen from Figure 2 that a 3D array is constructed where each plane along the third dimension of the array is a design structure matrix of the connections within that hyperedge set. In the bi-directional case, the intersection of all possible combinations of set elements are assigned a value of one for each hyperedge set present in the system.

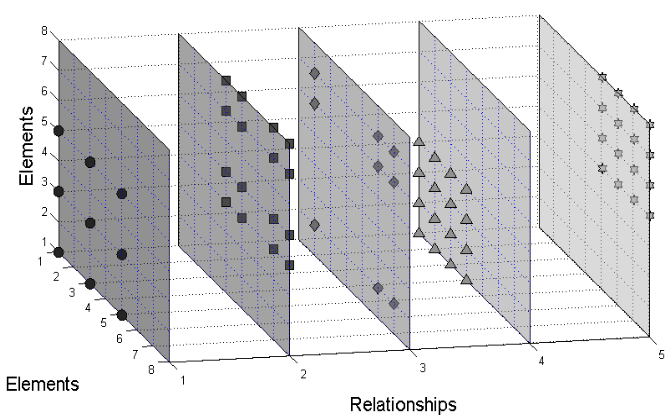

Figure 2: Translation of bi-partite graph to rDSM

This changes when the possibility of directional links is considered. This would be the case when modeling flows such as in electrical circuits, functional modeling, and manufacturing, among others. Given in Definition 2, the directional nature of the links is captured by dividing each relationship into sets of source and sink elements. The resulting rDSM can be considered to be source elements by sink elements by relationships. Directionality is modeled with the asymmetry of the matrices; a relation is found between source e_i to sink e_j but not found between source e_j to sink e_i. It should be noted that the diagonal is to be filled for all participating elements as elements are assumed to interact with themselves during any instance, such as a designer's thoughts during a meeting.

Definition 2

Given a set of elements $E = \{e_1, e_2, e_3, \ldots, e_{N_E}\}$ and a set of uni-directional relationships $R = \{r_1, r_2, r_3, \ldots, r_{N_R}\}$ where $r_i = \{\subseteq E_{source}, \subseteq E_{sink}\}$, there exists a $N_E \times N_E \times N_R$ array A where $A(r_{i,source}, r_{i,sink}, i) = 1$ for $i = 1{:}\,N_R$.

Each hyperedge represents a single interaction instance, rather than an interaction classification domain, such as meetings or workspaces, which the hyperedge may belong to. This differentiates the rDSM from existing expansions on the DSM method which address domains and classes.

In Figure 2, the connections within the bi-partite graph are assumed to represent bi-directional links as social interactions are not uni-directional. The resulting rDSM is symmetric according to Definition 1. However, the capability exists to capture uni-directional and mixed connections, such as in this case an email, according to Definition 2.

The ability of the rDSM to capture multiple paths between element pairs is seen when the matrix is collapsed into a single plane to form a multigraph projection of the rDSM hypergraph. This projection is achieved by summing the rDSM along the relationships dimension according to Equation 6.

$$[DOF] = \sum_k^{N_R} A_k \qquad (6)$$

The resulting matrix will no longer differentiate between distinct relationships and will not be a binary set of inter-relationships, but rather the number of edges linking each element pair. This can be considered to be the degree of freedom (DOF) matrix, as seen in Figure 3. The term Degree of Freedom in this case refers to the number of interaction parameters which are available for change, rather than referring to mechanical movement.

	Element							
Element	2	1	2	1	1	0	0	0
	1	2	1	1	0	1	1	0
	2	1	2	1	1	0	0	0
	1	1	1	2	1	0	1	1
	1	0	1	1	3	1	2	2
	0	1	0	0	1	2	2	1
	0	1	0	1	2	2	3	2
	0	0	0	1	2	1	2	2

Figure 3: Degree of Freedom Matrix

4 COMPLEXITY MEASUREMENT

The rDSM representation enables the measurement of several complexity metrics regarding size and interconnectivity. Size can be measured in terms of elements, relationships, and connections. Interconnectivity can be evaluated through degrees of freedom [7], all-pairs shortest path analysis [23], and the coupling complexity algorithm proposed by [10]. It has been argued that measuring complexity is critical to support informed comparisons between design problems, products, and processes [7].

4.1. Size Measurement

Size is the most common type of measurement used in complexity measurement today [10,7,24]. The size of any given object is based on the count of some classification of object within the system. It follows intuitively that if the number of elements or connections in a system increases, so does the system complexity. [25] This holds true for many different count-based metrics such as elements, relationships, connections, and classification types. However, while counts are the most intuitive form of complexity measurement it should be noted that their contribution to complexity is non-linear.[5] When the count is low, the addition of one more is significant, while the opposite is true of high-count systems. This can be modeled using information theory to define a number of bits present.[26]

4.1.1. Dimensional Size

Here, size measurements are taken by evaluating properties of the rDSM. Elements and relationships are defined through dimensional size. Elements represent the x-axis and y-axis size, while relationships represent the z-axis. For example, the rDSM presented in Figure 2 has eight elements and five relationships.

This results in three measures for the dimensional size of the system. The first two measures, as stated, are the number of relationships and elements within the system, given by Equations 7 and 8.

$$DS_R \stackrel{\text{def}}{=} N_R \qquad (7)$$

$$DS_E \stackrel{\text{def}}{=} N_E \qquad (8)$$

The third measure given is the volume of the system space defined by the rDSM. This volume represents the total number of connections that could exist within the system, excluding the area of self connection along the diagonal plane as defined by Equation 9.

$$DS_{ER} \stackrel{\text{def}}{=} N_E^2 N_R - N_E N_R \qquad (9)$$

4.1.2. Connective Size

The diagonal plane is addressed in measuring the connective size of the system. Connective size is defined by the number of arcs within the bipartite graph. This is measured by the sum of the degrees of the element nodes in the bipartite graph through the projected diagonal plane of the rDSM seen in the DOF multigraph projection as given by Equation 10.

$$CS = \sum_i^{N_E} \left[\sum_k^{N_R} A_{iik} \right] \qquad (10)$$

In our example case, we can refer back to Figure 3 to measure connective size and make observations about the system. Most of the designers participate in two relationships, one meeting and their workspace, while two designers participate in three relationships, indicating an additional meeting, for a connective size of 18. Additionally, it can be observed that the designers with more interactions are possibly leaders within the design group. These measures can be confirmed visually by reviewing the original bi-partite graph.

Degree of freedom serves as a compliment to connective size through a statistical approach. Degree of freedom refers to the number of parameters which may vary in the system. In the sense of the rDSM representation, this represents the number of element pairs which are connected through each of the relationship instances. Element pairs are evaluated irrespective of directionality in this measurement. As such, each relationship plane is transposed upon itself and the upper triangular taken one off the diagonal to cancel out directionality. The sum count of these upper triangular matrices is then taken for all non-zero element pairs as given be Equation 11.

$$DOF = \sum_k^{N_R} \sum_j^{N_E} \sum_i^{N_E} \left[U^{k=1}\left(A_{ijk} + A_{ijk}^T\right) \geq 1 \right] \qquad (11)$$

As the example case we are exploring here is purely bidirectional, the transpose and non-zero operations are not necessary to achieve the same results. For an entirely bidirectional case, the degree of freedom measurement can be found as the sum of the first diagonal upper triangular of the Degree of Freedom multigraph projection. If one refers back to Figure 3, it can be readily derived that the design group has 24 degrees of freedom, indicating 24 interaction parameters which are subject to change in the current system.

There exists a relationship between degree of freedom and connective size. If the system measured has no hyperedge relationships (each relationship contains only one connected element pair) then the system will have exactly one half the number of degrees of freedom as its connective size. This represents the minimum value for degree of freedom in any given system. It can be inferred then that the example case, having 24 degrees of freedom, represents a highly interconnected system with a high instance of hyperedge relationships. This is in fact the case, as all of the relationships identified in the case are hyperedges containing more than one connected element pair.

The size measurements for the example problem of design team meetings are found in Table 1. However, it should be noted that size is not sufficient for fully capturing complexity [10].

Table 1: Size Metrics

Class	Type	Metric	Design Group (Figure 1)
Size	Dimensional	Elements (DSE)	8
		Relationships (DSR)	5
	Connective	Connective Size (CS)	18
		Degree of Freedom (DOF)	24

4.2. Interconnectivity Measurement

The size of the system fails to capture the construction of the system. Consider a deck of cards in a stack and the same deck assembled into a house of cards. While both of these systems are of the same size in terms of elements and the set of exhaustive possible relationships between elements, the house of cards is clearly more complex. This complexity is derived from the interconnective structure formed by the house of cards. It is for this reason that properties of this structure must be taken into account when evaluating complexity.

Interconnectivity is measured through mathematical and algorithmic analyses applied to the rDSM and each represents a different aspect of the system complexity. Path length analysis evaluates the properties of information flow through the system and decomposability addresses how quickly the system may be disassembled in a systematic manner. These metrics complete the image of complexity.

4.2.1. Path Length

In this paper, we explore four specific path length based metrics. However, there are additional metrics that can be derived from the path length view of graphs, such as the longest path between nodes. These other metrics are deemed out of scope for this paper and are not considered here. Path length measurements are based on the number of relationships which must be passed through to travel from one element to another [9,23]. For example, to travel through the system A>B>C from A to C is a path length of 2. Here, we focus on the measurement of the shortest available path between any two elements in the system.

The measurement of shortest path length, like degree of freedom, is rooted in a matrix representation and the properties of this matrix. However, unlike degree of freedom, shortest path measurement is not derived from a projection of the rDSM but rather from a matrix resulting from an algorithmic treatment of the binary design structure matrix [17].

The design structure matrix is derived from the DOF multigraph projection by applying a logical test for all non-zero element pairs according to Equation 12.

$$[DSM] = \sum_{k}^{N_R} (A_k) \geq 1 \qquad (12)$$

This results in the classical binary DSM that is needed for shortest path evaluation. In the example case this yields the matrix shown in Figure 4. All-pairs shortest path analysis develops a matrix map of the number of relationships required to relate any two elements within the given system. This is achieved through an algorithmic set of matrix transforms performed computationally.

	Element							
Element		1	1	1	1	0	0	0
	1		1	1	0	1	1	0
	1	1		1	1	0	0	0
	1	1	1		1	0	1	1
	1	0	1	1		1	1	1
	0	1	0	0	1		1	1
	0	1	0	1	1	1		1
	0	0	0	1	1	1	1	

Figure 4: Design Structure Matrix

The design structure matrix is taken as the input to Algorithm 1, which is based on that developed by [27]. This algorithm transforms the DSM into a cube and adds the latter two dimensions of this cube together. The lowest values along the first dimension in the resulting array are then taken, resulting in a new matrix. This matrix is compared against the matrix which the iteration began with, once again taking the smallest values along the first dimension. This sequence is repeated until there is no change in the matrix from one iteration to the next.

Two post processing steps remove the matrix identity and infinite distances. The identity is considered to not be relevant to the measurement as the distance between an element and itself is, by definition, zero. Infinite distances are set to zero because the disconnection of elements is considered to be the least complex state possible as these elements will be incapable of manifesting higher order behavior between them.

Algorithm 1: Kelder All-Pairs Shortest Path[27]

Let

$$B = [DSM] \tag{13}$$

$$B_{B=0} = \infty \tag{14}$$

$$C_{1:N_E,1:N_E} = 1 \tag{15}$$

While $C \neq 0$, let

$$C = B \tag{16}$$

followed by

$$B = \min \left\{ B, \min_i \begin{pmatrix} \{B_{ij\,1}, B_{ij\,2}, \ldots, B_{ij\,N_E}\} \\ +\{B_{i1j}, B_{i2j}, \ldots, B_{iN_Ej}\} \end{pmatrix} \right\} \tag{17}$$

followed by

$$C = B - C \tag{18}$$

Then, let

$$\{B_{1,1}, B_{2,2}, \ldots, B_{N_E,N_E}\} = 0 \tag{19}$$

$$B_{B=\infty} = 0 \tag{20}$$

where

$$[ASP] = B \tag{21}$$

When this algorithm is applied to the design group example, the result is the matrix shown in Figure 5. The value shown in each cell represents the number of relationships which must be traversed for information to travel from one designer to another. While there are a considerable number of direct interactions, denoted by the number one, there are paths of length two in the system, particularly between the two workspaces.

Element	Element							
		1	1	1	1	2	2	2
	1		1	1	2	1	1	2
	1	1		1	1	2	2	2
	1	1	1		1	2	1	1
	1	2	1	1		1	1	1
	2	1	2	2	1		1	1
	2	1	2	1	1	1		1
	2	2	2	1	1	1	1	

Figure 5: All-pairs Shortest Path Matrix

An important observation which can be made in this case is that the path length results here contradict the conclusions based on connective size. While connective size would suggest that designers five and seven are the best connected to the rest of the group, the shortest path results suggest that it is in fact designers four and five which are best connected with only one other designer not directly connected to each of them. Designer four is actually the most efficiently connected member of the design group.

In addressing the complexity of the system, the first metric which develops from the all-pairs shortest path matrix is the total path length. This measurement, given by Equation 22, is the sum of the matrix and represents the total number of relationships traversed in travelling from each element to every other element.

$$TPL = \sum_i^{N_E} \sum_j^{N_E} B_{ij} \tag{22}$$

The total path length value is a combination of both a size and interconnectivity measurement due to its replication over all of the possible element pairs. This can be illustrated by comparing the example system's 24 degrees of freedom to its total path length of 74. While there are only 24 interaction parameters in the system, there are 74 unidirectional flows which these parameters influence. This is important as it is a measure of information exchange within the system.

Abstracting the total path length to remove the exponential size effect of elements yields the average shortest path length of the system. This is done by dividing the total path length by the size of the matrix minus the unused diagonal as shown in Equation 23. As a measurement neutral of elemental size, resulting value can be used to make a number of different determinations regarding the interconnective properties of the system.

$$APL = \frac{\sum_i^{N_E} \sum_j^{N_E} B_{ij}}{N_E^2 - N_E} \tag{23}$$

The average shortest path length represents the linearity of the system. A higher average shortest path length value will indicate a more linear system, while a lower value will indicate a more interconnected system. For a system without uncoupled components, the lowest possible average shortest path length will be one, representing a system in which all elements are directly connected to all other elements. An average shortest path length less than one can only occur if the system contains uncoupled components. The largest possible average shortest path length will occur when the system is purely linear.

For the example case, the average shortest path length is 1.3214 relationships per element pair. This indicates how close this system is to being fully interconnected, with many designers directly linked to one another. This level of interconnection is bolstered by the maximum shortest path length of the system, given by Equation 24.

$$MPL = \max_i \max_j B_{ij} \tag{24}$$

As the value of the maximum shortest path length must be an integer, it can be treated as a classification of the system. This classification represents the highest number of relationships which may be needed to relate any element to any other element. The example design group is of maximum shortest path length of two. This places this system in the lowest bracket of linearity short of being fully interconnected.

The final path length metric, shortest path length density is similar to that of average shortest path length but in this case applied to the broader rDSM representation as the size in question. As such, the formulation for shortest path length density multiplies the element pair size of the traditional DSM by the number of relationship planes in the rDSM representation as shown in Equation 25.

$$PLD = \frac{\sum_i^{N_E} \sum_j^{N_E} B_{ij}}{N_R\left(N_E^2 - N_E\right)} \qquad (25)$$

Shortest path length density serves as a test for the level of interconnection created on average by each relationship. For example, a system having only a single relationship connecting all elements will have an average shortest path length of one as well as a shortest path length density of one. If a second relationship is added to such as system, the shortest path length density falls to one half. As most systems will have many relationships, the typical shortest path length density will be very low such as in the case of the design group where the value is 0.2643.

The behavior of this measurement allows it to be considered as the interconnective efficiency of the system. Thus, the design group's interconnection is about 26% efficient. This addresses the use of multiple relationships to connect the same two elements. For example, it may not be necessary for two designers to attend the same two meetings or for designers who are already working together in the same workspace to have a meeting together at all.

4.2.2. Decomposability

The final measurement to be explored in this paper is that of decomposability. This addresses the steps which must be taken to disassemble the system in a structured manner. As a measure of complexity, the decomposability score increases with ever larger and more complex systems, thus what we are measuring is how difficult it is to take apart the system piece by piece.

It is the iterative reduction of the system which the Ameri-Summers decomposability algorithm [10] seeks to measure. Each step consists of removing those relationships that link to the elements with the fewest connections. Each additional step, relationship set, or relationships per separated element required to decompose the system is considered to increase the complexity.

Algorithm 2: Ameri-Summers Decomposability [10]

1. Eliminate Unary Relations
   (do not contribute to connectivity)
2. Initialize values: level = 1; total = 0;
3. For each graph to be searched
   a. Initialize set size = 1
   b. For all combinations of relations in a set size
      i. Remove set size relations from the graph
      ii. Check for separation
      iii. If separated graphs, mark the relation set removed
   c. If no relation set removed, increment set size and return to 3.b
   d. For all relation sets marked, find the combination of sets that remove the most relations without removal of elements of degree greater than set size (number of sets)
   e. Calculate score: level * set size * number of sets = total
   f. Submit each distinct graph to 3

Here, the Ameri-Summers decomposability algorithm has been amended slightly. This pertains to step 3.d and disallows the removal of a relationship if it will disconnect elements which have a degree higher than that of the removal set size. This corrects anomalies in the original algorithm which would lead to instances of the entire system being reduced in a single step due to auxiliary connections to the relationships to be removed beyond the least connected elements.

This would have been the case with the design group example presented here. The large number of elements connected to two relationships would result in all of the relationships being removed in a single step without capturing the two elements which are connected to three relationships.

The additional requirement proposed here changes this such that the first step will remove the first and fourth relations as indicated in Figure 6.

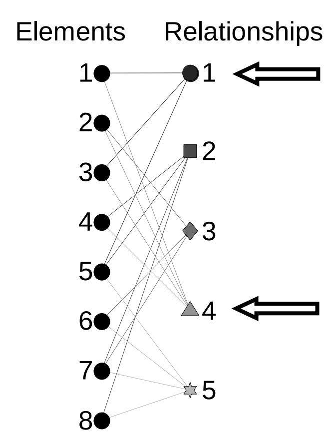

Figure 6: Ameri-Summers Decomposition Level 1

This removes the first and third elements as these elements shared the removed relationships as their only connections. Additional relationships are not removed as any other removal of size two would result in the elimination of elements which are not of the lowest degree.

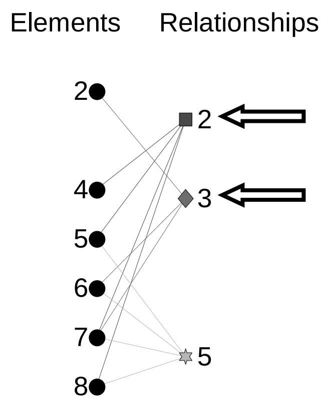

Figure 7: Ameri-Summers Decomposition Level 2

The results of the first level create the subgraph shown in Figure 7. Here, it can be seen that elements two and four are the least connected at degree one. Further, it can be seen that removing the relationships these elements are connected to, two and three, do not share an element that is connected to only these two relationships, therefore both may be removed in the second level of the decomposition. The third and final level of the decomposition now must only remove the fifth relationship to eliminate the remaining elements as shown in Figure 8

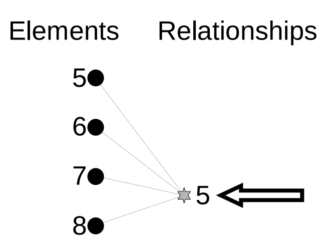

Figure 8: Ameri-Summers Decomposition Level 3

Decomposability value generated in this process is calculated as defined in Algorithm 2. The first level removes a single set of two for a value of two. The second level removes two sets of one for a value of four. The third level removes a single set of one for a value of three. Therefore, the Ameri-Summers value for this system is nine. While this value does not have physical meaning to the system as the other metrics which have been presented do, it does serve as a tool for comparison between different systems.

The interconnectivity measurements for the example problem of design team meetings are found in Table 3.

Table 2: Interconnectivity Metrics

Class	Type	Metric	Design Group (Figure 1)
Interconnection	Path Length	Total Path Length (TPL)	74
		Maximum Path Length (MPL)	2
		Average Path Length (APL)	1.3214
		Path Length Density (PLD)	0.2643
	Decomp.	Ameri-Summers (ASA)	9

5 MEASUREMENT AGGREGATION AND APPLICATION

The comparison of systems is the basis behind the aggregate consideration of complexity measurements. As each metric presented here behaves in a unique manner and scale, the value for each metric may be desired to be larger or smaller depending on the type of system analyzed and the design goals being addressed. Therefore the calculation of a single value for the complexity of the system would be counter intuitive and of very little use. Rather, metrics are to be considered as a group based on the properties and goals related to each in order to make comparisons between systems.

5.1. Function Structure Example

There are several approaches to comparing systems. In order to demonstrate these, we will depart from the design group example and compare the function structures of two consumer products, a sander shown in Figure 9 and an electric screwdriver shown in Figure 10. These function structures were selected from those available in the Design Repository[1]. In the modeling of these systems, the functions were considered to be elements and the flows considered relationships. An additional node was added to each system to represent the environment in order to capture the inputs and outputs. It should be noted that these systems carry direction, unlike the design group example.

[1] http://function2.mime.oregonstate.edu accessed 2010.02.04

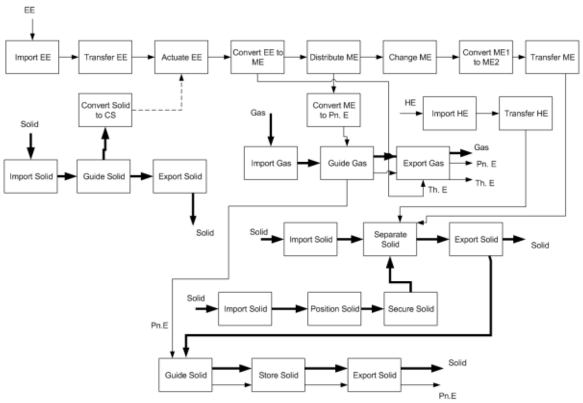

Figure 9: Sander Function Structure

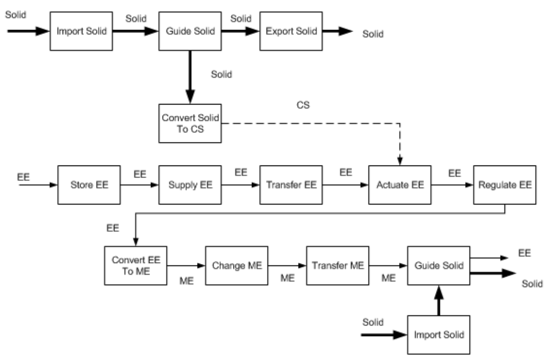

Figure 10: Electric Screwdriver Function Structure

5.2. Aggregation Approaches

The first of the aggregation approaches is the basic set of metric values. Comparing two sets of values pertaining to the properties of different systems is common practice in engineering discipline, such as in comparing the specifications of potential components. This is, however, a somewhat cumbersome approach at times, compounded by the varying scales and behaviors of the different metrics. For example, the function structures may be compared as in Table 3. Here, it is possible to see a general trend towards the sander being more complex, but it is difficult to grasp the extent.

Table 3: Set Based Comparison of System Complexity

Class	Type	Metric	Sander	Screwdriver
Size	Dimensional	Elements (DSE)	28	15
		Relationships (DSR)	43	19
	Connective	Connective Size (CS)	86	38
		Degree of Freedom (DOF)	47	19
Interconnection	Path Length	Total Path Length (TPL)	2294	602
		Maximum Path Length (MPL)	6	5
		Average Path Length (APL)	3.0344	2.8667
		Path Length Density (PLD)	0.0706	0.1509
	Decomp.	Ameri-Summers (ASA)	129	24

The next approach is to arrange the metrics as the components of a 9-dimensional vector. This enables the comparison of systems based on Euclidian distance and other vector operations. However, this approach remains susceptible to the varied scales of the metrics. For example, the base Euclidian distance between the sander and screwdriver would be 1696.3, largely due to the total path length. To counteract this, each metric is normalized to the largest value for that metric across all of the systems to be compared. This results in a Euclidian distance of 1.63. With all values falling between zero and one, the metrics can then be compared on an equal footing and suggesting how different the complexity of one system is from another.

The final approach is an extension to the normalized vector approach to visualize the vector elements in terms of the goals for each metric. This may be done in two ways. One way is to normalize metrics against desired values, creating a target value of one for all metrics. This requires a level of knowledge regarding what these target values should be which is not presently available. However, the other method is more straight forward in that the values may be subtracted from one if a larger value is considered desirable or left as is if a smaller value is better.

5.3. Spider Graph Comparison

Such a prioritized vector can then be visualized in a spider, or radar, graph. This graph presents each vector component as an axis, with all axes radiating from a common origin at equal intervals. The value of each metric is plotted as a point on its axis and these points are joined to form a filled area for each system.

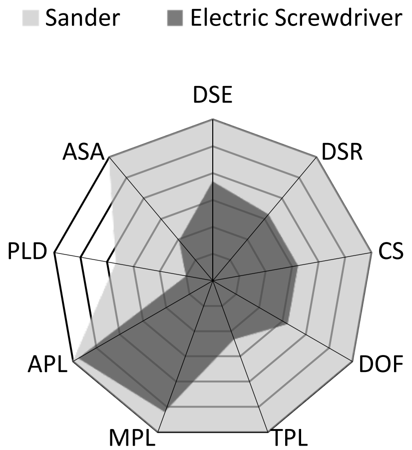

Figure 11: Spider Graph Comparison of Sander and Electric Screwdriver

For example, Figure 11 shows the spider graph for the sander and electric screwdriver. Here, it has been decided that a higher shortest path length density is desirable and therefore should be shown as less complex by subtracting the normalized values from one. The resulting figure clearly shows the electric screwdriver to be much less complex than the sander, particularly in regards to system size and decomposability. However, this also quickly highlights that, despite the smaller system size and easier decomposition, the shortest path lengths through the system are not significantly reduced. This is not surprising as these are both power tools which, though they perform different tasks, operate in a very similar manner.

This method of comparing the complexity of systems is much faster and more robust than the first approach of comparing sets of values. As more systems are added to the comparison, greater detail will be displayed in the spider graph. However it should be noted that the stacked view shown in Figure 11 cannot be used to compare more than two or three systems at a time. Additional systems will have to be plotted separately in order to be visible.

To illustrate the comparison of multiple systems, we will return to the design group example of Figure 1. A comparison is made between the original construct of the design group's activities and two proposed alternatives. Assuming that workspaces are unchangeable, the occurrence and attendance of meetings are altered.

Meeting number two between designers four, five, seven, and eight connects both of the designers who participate in more than one meeting. Therefore, it is proposed that this meeting be eliminated from the schedule. However, this would result in designers four and eight having no meetings at all. So, a second alternative is proposed in which two designers from each workspace attend one meeting and the other two attend a different meeting.

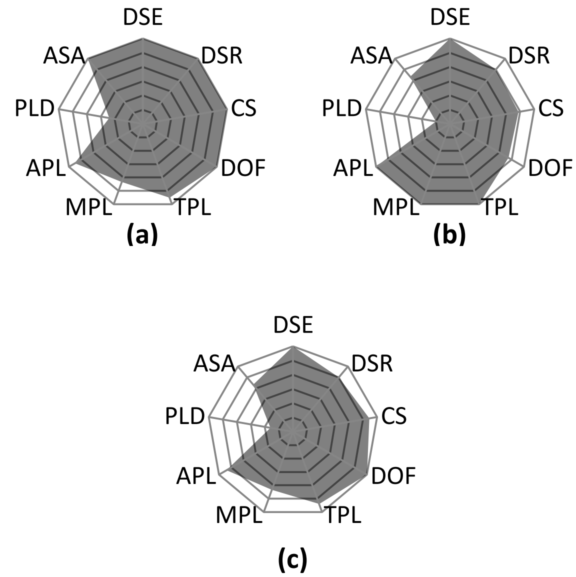

Figure 12: Spider Graph Comparison of (a) Original Design Group, (b) Removal of Meeting 2, and (c) Balanced Meetings

When the resulting metric values of these systems are compared the resulting spider graphs are those shown in Figure 12. Here, the assumption that higher shortest path length density is less complex has been carried over from the function structure example.

It can be seen that the cancellation of the second meeting does indeed reduce the complexity of the system, particularly in regards to the size class. However, the path lengths in the system increase as a result of some designers not attending meetings. It could be considered that the increases in path length may be offset by the reduction in size and the improvement of path length density, until the third option is considered.

The third option, in which two meetings of balanced attendance are scheduled, achieves the benefits of the second meeting being canceled without the negatives. The size of the system is reduced and the shortest path length density is improved while the path lengths are themselves either reduced or unchanged. Thus, it can be concluded that the third option is the preferable arrangement for the design group of the options considered here. This is an example of how this method can be used to quickly assess proposed system variations for their impact on complexity.

6 CONCLUSIONS AND PATH FORWARD

We have established in this paper that complexity is the effort required to understand a given system and that this effort is based on a collection of attributes rather than a single value. These attributes are divided into classes of size and interconnectivity. Size is in turn defined in terms of dimensionality and connectivity in the system. Interconnectivity addresses the structural arrangement of the system through shortest path length analysis and systematic decomposition.

These attributes of complexity are presented in a normalized graphical depiction for comparison of systems. The systems compared are normalized against the highest value presented for any given attribute and adjusted to reflect priorities for the given system type. The graphical depiction takes the form of a spider graph which presents each normalized attribute on an independent axis. This provides a quick approach to visually comparing the complexity of systems.

However, there are several areas which remain to be addressed. Chief among these is the need for an empirical evaluation for the interpretability of the single visualization tool presented here. User studies may be conducted to fill this gap.

Additional studies may also be used to establish whether these sets of metrics can be used to predict the performance of the systems modeled. An example of this would be utilizing these metrics to predict the cost or development time of a consumer product. This may be achieved through the direct relation of metrics to performance or the application of metrics as properties of distribution and trend curves.

Some opportunity exists for improvement of the metrics, particularly with regard to path lengths and system domains. Path lengths have the potential for extension through the application of distances to what are presently only binary connections. This would allow for the capture of connections of varying strength. The approach here has thus far been restricted to elements of a single domain. An exploration of how this may be applied to multiple domains and the connections between those domains presents another possible avenue of investigation.

REFERENCES

[1] N.P. Suh, "Axiomatic Design Theory for Systems," *Research in Engineering Design*, vol. 10, no. 4, pp. 189-209, 1998.

[2] G. Pahl and W. Beitz, *Engineering Design: A Systematic Approach*, 3rd ed. London: Springer, 2007.

[3] G. A. Hazelrigg, *Systems Engineering: An Approach to Information-Based Design*, 1st ed. New Jersey, NJ, USA: Prentise Hall Inc, 1996.

[4] H. A. Simon, *The Sciences of the Artificial*. Cambridge, MA, USA: The MIT Press, 1981.

[5] I. Barclay and Z. Dann, "New-product-development performance evaluation: a product-compexity-based methodology," *IEE Proceedings in Scientific Measurement Technology*, vol. 147, no. 2, pp. 41-55, march 2000.

[6] J. Casti, *Connectivity, Complexity, and Catastrophe in Large-Scale Systems*, International Series on Applied Systems Analysis ed. New York, NY, USA: John Wiley & Sons, 1979.

[7] J.D. Summers and J.J. Shah, "Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability," *Journal of Mechanical Design*, vol. 132, no. 2, p. 021004 (11 Pages), February 2010.

[8] L. Lofgren, "Complexity of Descriptions of Systems," *International Journal of General Systems*, vol. 3, pp. 197-214, 1977.

[9] J. Mathieson and J. Summers, "Relational DSMs in Connectivity Compexity Measurement," , Greenville, SC, 2009.

[10] F Ameri, J.D. Summers, G.M. Mocko, and M. Porter, "Engineering design complexity: an investigation of methods and measures," *Research in Engineering Design*, vol. 19, no. 2-3, pp. 161-179, November 2008.

[11] J. Ahn and R. Crawford, "Complexity Analysis of Computational Engineering Design Processes," in *Design Engineering Technical Conferences - Design Theory and Methodology*, New York, 1994, pp. 205-220.

[12] R. Sedgewick, *Algorithms in c++*. Menlo Park, CA, USA: Addison-Wesley, 1998.

[13] D. Varma and E. Trachtenberg, "On the Estimation of Logic Complexity for Design Automation Applications," in *International Conference on Computer Design: VSLI in Computers and Processors, IEEE*, Cambridge, MA, 1990.

[14] D. Braha and O. Maimon, "The Measurement of a Design Structural and Functional Complexity," *IEEE Trans. Syst.Man Cybern., Part A. Syst. Humans*, vol. 28, no. 4, pp. 116-132, 1998.

[15] B. El-Haik and K. Yang, "The Components of Complexity in Engineering Design," *IIE Trans.* , vol. 31, pp. 925-934, 1999.

[16] A. Kolmogorov, "Combinatorial Foundations of Information Theory and Calculus of Probabilities," *Russ. Math. Surveys*, vol. 38, pp. 29-40, 1983.

[17] D.V. Steward, *Systems Analysis and Management: Structure, Strategy and Design*. New York: Petrocelli Books, Inc., 1981.

[18] M. Maurer, *Structural Awareness in Complex Product Design*. Munchen, Germany: FM-Kopierbar, 2007.

[19] J. D. Summers, B. Bettig, and J. J. Shah, "The design exemplar: a new data structure for embodiment design automation," *Journal of Mechanical Design*, vol. 126, no. 5, pp. 775-787, September 2004.

[20] M. Kreimeyer, S., Gurtler, M. Braun, and U. Lindemann, "Relating Two Domains via a Third - An Approach to OVercome Ambiguous Attributions using Multiple-Domain Matrices," in *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2008)* , Brooklyn, New York, USA , 2008, pp. 297-306.

[21] M.E.J. Newman, "The structure and function of complex

networks," *SIAM Review*, pp. 167-256, 2003.

[22] Rene Keller, Claudia Eckert, and P. John Clarkson, "Matrices or Node-Link Diagrams: Which Visual Representation is Better for Visualizing Connectivity Models," *Information Visualization*, vol. 5, pp. 62-76, 2006.

[23] I. Pramanick and H. Ali, "Analysis and experiments for a parallel solution to the all pairs shortest path problem," in *1994 IEEE International Symposium on Circuits and Systems*, vol. 1, New York, NY, USA, 1994, pp. 479-82.

[24] James L. Mathieson, Chiradeep Sen, and Joshua D. Summers, "Information Generation in the Design Process," in *Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, San Diego, CA, USA, 2009, pp. 1-10.

[25] Anthony J. Masalonis, Michael B. Callaham, and Craig R. Wanke, "Dynamic Density and Complexity Metrics for Realtime Traffic Flow Management," Center for Advanced Aviation System Development (CAASD), The MITRE Corporation, McLean, VA, USA, 2003.

[26] Abhijit V. Deshmukh, Joseph J. Talavage, and Moshe M. Barash, "Complexity in Manufacturing Systems Part I: Analysis of Static Complexity," Unpublished.

[27] Michael Kleder. (2007, December) "All-pairs Shortest Path" Graph Solver. Open-source MATLAB Code.

