MIT Open Access Articles

Structural Complexity Quantification for Engineered Complex Systems and Implications on System Architecture and Design

Massachusetts Institute of Technology

DETC2013-12013

STRUCTURAL COMPLEXITY QUANTIFICATION FOR ENGINEERED COMPLEX SYSTEMS AND IMPLICATIONS ON SYSTEM ARCHITECTURE AND DESIGN

Kaushik Sinha
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139

Olivier L. de Weck
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139

ABSTRACT

The complexity of today's highly engineered products is rooted in the interwoven architecture defined by its components and their interactions. Such structures can be viewed as the adjacency matrix of the associated dependency network representing the product architecture. To evaluate a complex system or to compare it to other systems, numerical assessment of its structural complexity is essential. In this paper, we develop a quantitative measure for structural complexity and apply the same to real-world engineered systems like gas turbine engines. It is observed that low topological complexity implies centralized architectures and that higher levels of complexity generally indicate highly distributed architectures. We posit that the development cost varies non-linearly with structural complexity. Empirical evidence of such behavior is presented from the literature and preliminary results from simple experiments involving assembly of simple structures further strengthens our hypothesis. We demonstrate that structural complexity and modularity are not necessarily negatively correlated using a simple example. We further discuss distribution of complexity across the system architecture and its strategic implications for system development efforts.

Keywords - Structural Complexity, topological complexity, gas turbine engine, development effort, metric validation, complexity vs. modularity, complexity distribution, design encapsulation.

1. INTRODUCTION

Today's large-scale engineered systems are becoming increasingly complex due to numerous reasons including increasing demands on performance, and improved lifecycle properties. As a consequence, large product development projects are becoming increasingly challenging and are falling behind in terms of schedule and cost performance. For example, in 13 aerospace projects reviewed by the US Government Accountability Office (GAO) since 2008, large development cost growth of about 55% was observed. Such large development cost overruns/failures of large-scale system development projects can largely be attributed to our current inability to characterize, quantify and manage associated complexity [DARPA report, 2011]. With increasing complexity of engineered systems, typically the associated Life Cycle Cost (LCC) also increases [Sheard and Mostashari, 2010]. The challenge of quantifying and managing complexity is also central to many research areas occupied with engineering. A particular concern with the work done in the area of complexity estimation is that less than one-fifth of the studies even attempted to provide some degree of objective quantification of complexity [Tang and Salminen, 2001]. An objective and quantifiable measure of structural complexity is imperative for systematic search and optimization of system architecture. In particular, the consideration of the dependency network attracts attention in various scientific works because dependency-based system structures affect system characteristics and behavior. A system consisting of many components that are linked to each other and the interaction between these parts influences the system's behavior [Ulrich 1995, Lindemann et al. 2008]. The complexity of technical systems depends on the heterogeneity and quantity of different elements and their connectivity pattern, and is a measurable system characteristic. This internal product architecture can be represented by complex networks, which are a graph-theoretic representation of complex systems. The nodes, representing components of the systems, are connected by links if there exists a direct interaction between any pair of components [Sheard and Mostashari, 2010]. The product functionality is enabled by the underlying architecture. A perpetually occurring theme is the complexity of product architecture. It is often perceived that as we stretch the limits of efficiency and attempt to design more robust systems, we tend to make architectures more complex. In this paper, a rigorous

and quantitative structural complexity metric for architecture evaluation and optimization, incorporating the fundamental underlying characteristics of product architecture, is proposed. This lends objectivity to the process of product architecture selection and design. Subsequently this measure is applied to a pair of jet engine architectures to measure and compare their structural complexities. Next, we posit that the development cost varies non-linearly with structural complexity. Some empirical evidence of such behavior is present in the literature. This hypothesis is further buttressed by preliminary results from simple experiments involving assembly of simple molecular structures. Subsequently, we demonstrate the trade-off between structural complexity and modularity that shows complexity and modularity are not necessarily negatively correlated. We introduce the notion of structural complexity distribution across the system architecture and how this impacts strategic decisions in system development efforts.

NOMENCLATURE

α_i = complexity of ith component
β_{ij} = interface complexity between i^{th} and j^{th} component
n = number of components in a system
m = number of pair-wise interactions in a system
A = adjacency matrix of simple graph G with n nodes
$A^{(k)}$ = adjacency matrix of kth vertex-deleted subgraph of G
A_i = adjacency matrix of ith module or subsystem
$E(A)$ = matrix energy of A (i.e., sum of its singular values)
$\sigma_i(A)$ = i^{th} singular value of A
Δ = integrative topological complexity
γ = normalization factor
f_{ij} = interface-type complexity factor (ICF)
TRL = **T**echnology **R**eadiness **L**evel
θ = reconstructability coefficient
Q = modularity index

2. STRUCTURAL COMPLEXITY QUANTIFICATION

The structural complexity of technical systems depends on the quantity of different elements and their connectivity structure and is a measurable system characteristic. This representation includes contributions coming from the internal complexities of the components of the system; the complexities associated to the pair-wise connections among the components and a quantity that encapsulates the complexity due to inherent arrangement of connections (i.e., structure) amongst the components. We propose the following functional form for estimating the structural complexity of an engineered complex system:

Structural Complexity, $C = C_1 + C_2 C_3$

The first term C_1 represents the sum of complexities of individual components alone (local effect) and does not involve architectural information. The second term has two factors: (i) number and complexity of each pair-wise interaction, C_2 (local effect) and (ii) effect of architecture or the arrangement of the interfaces C_3 (global effect). Now, given the same number of

interfaces these can be arranged in a variety of patterns and the number of interfaces alone does not dictate how they should be arranged among themselves, given there are no additional system constraints. Hence in this sense, C_2 and C_3 are mutually independent, therefore the multiplicative model. The effect of system architecture captured in C_3 represents a global effect whose impact is typically first realized at the time of system integration. Similar functional forms are found in quantum mechanical analysis of molecular systems where the system Hamiltonian is the matrix of importance [Gutman et al. 1998, Sinha and de Weck 2012]. Let us look in detail at the topological complexity quantifier and then get at the structural complexity metric in fully expanded form.

2.1 TOPOLOGICAL COMPLEXITY METRIC

The topological complexity is defined as the matrix energy or graph energy of the adjacency matrix. Topological complexity originates from interaction between elements and depends on the nature of such connectivity structure. The adjacency matrix $A \in M_{nxn}$ of a network is defined as follows:

$$A_{ij} = \begin{cases} 1 \ \forall [(i,j) \,|\, (i \neq j) \text{ and } (i,j) \in \Lambda] \\ 0 \text{ otherwise} \end{cases}$$

where Λ represents the set of connected nodes and $\boldsymbol{n}$ being the number of components in the system. The diagonal elements of A are zero. The associated *matrix energy* of the network is defined as the sum of singular values of the adjacency matrix:

$$E(A) = \sum_{i=1}^{n} \sigma_i, \text{ where } \sigma_i \text{ represents } i^{th} \text{ singular value}$$

The *matrix energy* also expresses the minimal effective dimension embedded within the connectivity pattern represented through the binary adjacency matrix. Notionally, this quantity encapsulates the "intricateness" of structural dependency among components. Using singular value decomposition (SVD), we can express matrix A as:

$$A = \sum_{i=1}^{n} \sigma_i \underbrace{u_i v_i^T}_{E_i} = \sum_{i=1}^{n} \sigma_i E_i$$

where E_i represents simple, building block matrices of unit matrix energy and unit norm. Using this view, we observe that *matrix energy* or *graph energy* express the sum of weights associated with the building block matrices required to represent or reconstruct the adjacency matrix $\boldsymbol{A}$. This naturally leads us to the *graph reconstructability* viewpoint [Liu et al. 2010, Mieghem 2011] and matrix energy can be shown to have a dual behavior with respect to the network *reconstructability* coefficient θ [Liu et al. 2010]. We can view the ability to easily construct system structure as the dual of topological complexity. Minimum topological complexity mandates maximization of re-constructability (see fig. 1). This behavior will be further explored in the future.
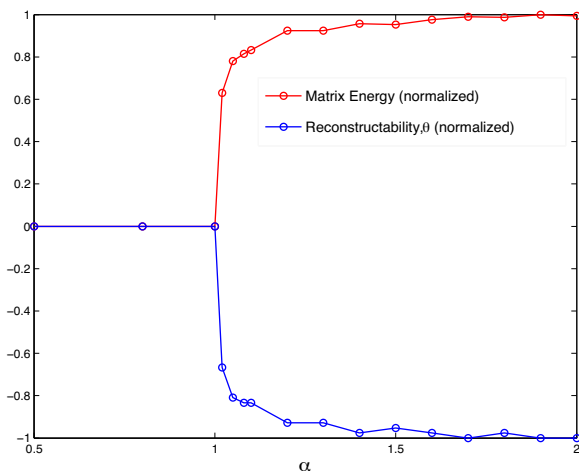
2

Copyright © 2013 by ASME

Figure 1. Observed *Dualism* between topological complexity metric (i.e., matrix energy) and reconstructability [12] on *Fabrikant* networks [2] with varying α. The two quantities are normalized in [0,1] (for matrix energy) and [-1,0] (for reconstructability coefficient) respectively for visualization.

Matrix energy is used as a measure of topological complexity of the system architecture and is invariant to isomorphic transformations of the matrix [Horn and Johnson 1994, Gutman et al. 1998, Nikiforov 2007, Mieghem 2011]. To check conceptual validity of the proposed topological complexity metric, we benchmarked this against the set of minimal required properties prescribed as Weyuker's criteria [Weyuker 1988] and also compared against some other complexity metrics that have been proposed in the existing literature (see table 1) [Lindemann et al. 2008]. As we can observe, the proposed topological complexity metric is fully compliant with Weyuker's criteria [Weyuker 1988].

Table 1: Benchmarking of matrix energy or graph energy against Weyuker's criteria (i.e., a set of nine criterion [Weyuker 1988]) and comparison with other proposed metrics of complexity.

Metric \ WC	1	2	3	4	5	6	7	8	9
Graph Diameter	✓	✓	✓	✓	✗	✗	✗	✓	✗
Degree of non-planarity	✓	✓	✓	✓	✗	✗	✓	✓	✗
Path Length	✓	✓	✓	✓	✗	✗	✗	✓	✗
Nesting Depth	✓	✓	✓	✓	✗	✗	✗	✓	✗
Graph Energy	✓	✓	✓	✓	✓	✓	✓	✓	✓

A more distributed system cannot be condensed/reduced significantly and such a system indicates higher structural complexity but it might also help achieve higher performance levels with high robustness and reliability. Topological complexity increases from *centralized* towards more *distributed* architectures (see fig. 2). Topological complexity helps distinguish structural complexity of two very different connectivity structures with same number of components and interactions (see fig. 3). For example, if we were to only consider the number of components and number of interactions as measure of complexity without considering the underlying

architecture, we would have thought that both the star architecture and the hierarchical tree architecture have the same structural complexity. This is an inherent problem with all simple counting based measures of complexity since they are incomplete when it comes to capturing global effects like system structure.
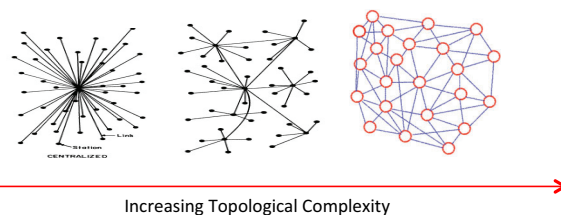

Figure 2. Evolution of topological complexity based on their internal structure: (a) 'centralized' or bus architecture; (b) hierarchical architecture and (c) 'distributed' architecture.
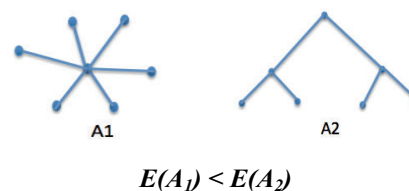

$$E(A_1) < E(A_2)$$

Figure 3. Two architectures having the same number of nodes and connections but are differentiated based on their internal structure with $E(A_1)$ = 4.9 and $E(A_2)$ = 6.83.

The topological complexity term captures the challenges associated with system integration [Sinha and de Weck 2012].

2.2 STRUCTURAL COMPLEXITY METRIC

The proposed structural complexity metric is defined below:

$$C = C_1 + C_2 C_3$$
$$= \sum_{i=1}^{n} \alpha_i + \left[\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} A_{ij} \right] \gamma E(A)$$

The implication of the different terms in the structural complexity metric can be found in [Sinha and de Weck 2012]. The individual component complexities can vary across the system (e.g., a low-pressure turbine is much more complex than the exhaust nozzle in a jet engine) and designated by α's, the Component Complexity Estimate (CCE). This measure could be based on the widely used notion of component TRL (i.e., Technology Readiness Level) or other similarly motivated measures as a surrogate for component complexity. We propose a component complexity scale of [0, 5] and computed from component TRL [Sadin *et al*., 1988] level as:

3   Copyright © 2013 by ASME

$$\alpha = 5\left(\frac{TRL_{\max} - TRL}{TRL_{\max} - TRL_{\min}}\right)$$

We describe here a multiplicative model for representing interface complexity β_{ij}. Each interface complexity depends on the complexities of the pair-wise interfacing components (α_i and α_j) and a coefficient characteristic of the interface type (f_{ij}):

$$\beta_{ij} = f_{ij}\alpha_i\alpha_j \ where \ \alpha_i, \alpha_j \neq 0$$

If there are multiple types of connections between two components (say, load-transfer, material flow and control action flow), the interface will have a high β value since it will be more 'complex' to achieve/design this connection compared to a simpler load-transfer connection. For large, engineered complex systems, it appears that β in [0,1] is a good initial estimate for the chosen component complexity range of [0, 5]. We are in the process of developing further guidelines for generating good estimates for f_{ij} for different interface types.

The topological complexity term scales with the challenges associated with system integration. Higher topological complexity will likely lengthen system integration efforts significantly and it is a global property that is not visible locally (i.e., during component or interface development). The implication of different terms of the structural complexity measure is described in the following diagram:
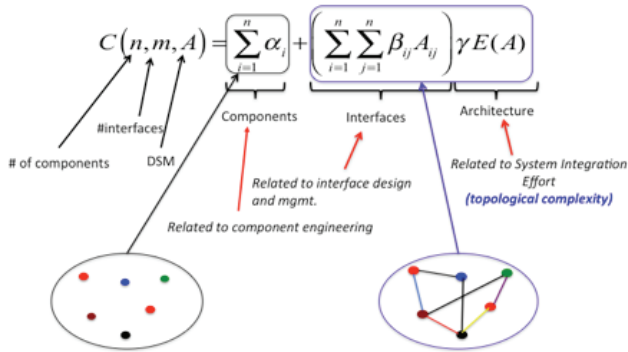

Figure 4. Constituents of different parts of the overall structural complexity metric and their implications in the context of system development. DSM stands for Design Structure Matrix [Ulrich and Eppinger, 1995].

We demonstrate operationalization of the proposed methodology using a simple example in the following sub-section and also briefly explain a larger jet engine example where this methodology has been applied and reported in an earlier article [Sinha and de Weck 2012]. This formulation can be used to show that structural complexity and modularity measure different aspects of the system architecture and need not necessarily be negatively correlated as often implied in the existing literature (more complexity does not automatically lead to less modularity and vice-versa). Consideration of system components and the underlying architecture also opens up an area where component-level and architecture-level complexities can be traded against each other (i.e., simple components with complex architecture vs. complex components with simpler architecture).

2.3 ILLUSTRATIVE EXAMPLES

We first present a small example of a hypothetical system for demonstrating the mechanics of the method. Let us start with the example of a hypothetical system representing fluid flow as shown in fig. 5.
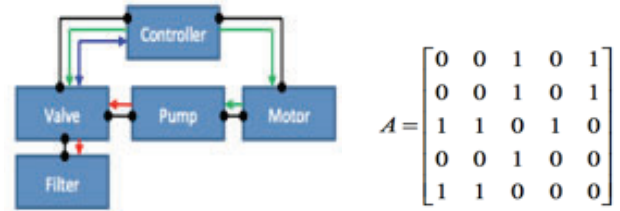

Figure 5. (a) Sample system: it shows different connection types amongst components - physical/mechanical connection (black); material/fluid flow (red); energy flow (green) and information/control signal flow (blue); (b) aggregated adjacency matrix. The labeling of components is as follows: Controller (1), Pump (2), Valve (3), Filter (40 and Motor (5).

The graph energy of this system is $E(A) = 5.6$. If we simply assume unit complexities for all components and connections, we have $C_1 = 5$ and $C_2 = 10$. If we use $\gamma = 1/n = 1/5$, the structural complexity is $C = (5+11.2) = 16.2$. Now let us differentiate among components and let the component complexity vector be {(controller=5); (pump=2);(valve=1); (filter=1);(motor=3)}. The sum of component complexities is 12. Let us use the following connection complexities: $\beta^{mech} = 0.5$, $\beta^{flow/energy} = 0.5$ and $\beta^{info} = 1.0$. Now the sum of connection complexities is $C_2 = 12$. The new structural complexity is thus 25.44. Here, the system components are more complex than the connection complexities and it has the effect of increasing the contribution of component complexities in the overall structural complexity metric. In practice, assignment of component and connection complexities could be uncertain during the conceptual design stage or even after the product architecture is finalized. In such cases, the resulting structural complexity will not be a single number but a distribution, depending upon the distribution of individual component and connection complexities.

The same method was applied to two different jet engine architectures, namely a dual spool direct-drive turbofan (e.g., older architecture) and a geared turbofan engine (e.g., new architecture). The specific details can be found in [Sinha and de Weck 2012]. The component complexities were assessed by experts using a scale of [0,5], while all connections were assumed to be of the same complexity due to lack of available data. The experts in this study were engineers involved in the development of the aircraft engines.

Table 2: Comparison of dual spool direct-drive turbofan (i.e., older architecture) and geared turbofan (i.e., new architecture) architectures and their structural complexities.

4   Copyright © 2013 by ASME

Architectural attribute	Older Architecture	New Architecture	Change
Components	69	73	6%
Components + Connections	69 + 269 = 338	73 + 361 = 434	28%
Topological Complexity	104.4	123.3	18%
Structural Complexity	548	767	40%

It was found that the structural complexity increase is grossly underestimated by considering only the number of connections and pair-wise interfaces by 43% (e.g., a 40 % increase in complexity vs. a 28% increase predicted by considering only components and pair-wise interfaces). This example manifests the importance of system architecture and its influence on the structural complexity. This comparison is of interest because of the potential benefits of the geared turbofan architecture in terms of various performance measures (i.e., fuel burn, noise). The development cost of the geared turbofan is much higher than the previous generation engine and we believe increased structural complexity is one of the primary contributors for this behavior [Denman et al. 2011, Sinha and de Weck 2012].

2.4 SENSITIVITY ANALYSIS

Sensitivity analysis with respect to individual component complexities is relatively straightforward as it follows the parametric sensitivity analysis procedure since the underlying system architecture remains unchanged. Here we look at the sensitivity of component deletion on the structural complexity metric. This has a combinatorial effect as the underlying system architecture is changed.

Let us consider a system architecture be represented as a simple graph G with n components and m interactions and whose binary adjacency matrix is A. Now the kth component (i.e., k^{th} node of graph G) is removed from the system and this results in deletion of all interactions associated to this component. For simplicity, assume that such deletions do not render the overall system totally dysfunctional (but performance might partially degrade) [Agte *et al.*, 2012]. In case of component removal leading to structural disintegration of the system (i.e., system fragments into multiple disconnected fragments), the sensitivity for that component is set to a very large number, indicative of this behavior. We can also employ the same strategy for absolutely necessary components of the system (i.e., components that cannot be removed else the system becomes *dysfunctional*) from system functionality stand point.

We also assume that component deletion does not result in any re-distribution of component and interface complexities while maintaining at least limited functionality. Please note that we are not imposing the multiplicative model for estimating interface complexity here. Imposing the multiplicative model would result in a slightly different mathematical expression, but essential characteristics remain the same.

Under these assumptions, we can express the difference in structural complexity due to removal of kth system component as below:

$$\Delta C = (C_1 - C_1^{(k)}) + C_2 C_3 \left[1 - \frac{C_2^{(k)} C_3^{(k)}}{C_2 C_3} \right]$$

$$= \underbrace{\left(\sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n-1} \alpha_i \right)}_{\substack{\text{effect of individual} \\ \text{component}}}$$

$$+ \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} \right) \frac{E(A)}{n} \left[1 - \underbrace{\left(1 - \frac{\sum_{i=1}^{n} \beta_{i,k} + \sum_{i=1}^{n} \beta_{k,i}}{\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}} \right) \left(\frac{n}{n-1} \right) \underbrace{\left(\frac{E(A^{(k)})}{E(A)} \right)}_{\text{topological effect}}}_{\text{effect of deleted interfaces}} \right]$$

$$= \alpha_k + \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} \right) \frac{E(A)}{n} \left[1 - \underbrace{\left(1 - \frac{\sum_{i=1}^{n} \beta_{i,k} + \sum_{i=1}^{n} \beta_{k,i}}{\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}} \right) \left(\frac{n}{n-1} \right) \underbrace{\left(\frac{E(A^{(k)})}{E(A)} \right)}_{\text{topological effect}}}_{\text{effect of deleted interfaces}} \right]$$

Looking at the different terms of the above expression, we observe that for any component, sensitivity to its deletion on structural complexity consists of three sources: (i) complexity of the deleted component itself, (ii) complexities of the deleted interactions that were associated with the removed component, and (iii) re-structuring of the underlying system architecture due to removal of the kth component. The only impact that organization of system elements has on the sensitivity expression is through the changes in topological complexity term after removal of any system component. If removal of any component makes the system structure less distributed than before, then the ratio of topological complexities in the above expression becomes much larger than unity. This methodology was applied to different components of the jet engine example presented in the previous section.

The detailed sensitivity analysis, reveals that primary functionality generators (e.g., those generating thrust) are significant contributors to component complexity while supporting systems (e.g., lubrication systems, accessory gearbox, engine control systems) are the primary contributors to topological complexity and have significant impact on system integration efforts [Denman *et al.* 2011, Sinha and de Weck 2012]. This also showed a very interesting scenario where an *Air Buffer Cooler*, a very simple component, could have a more significant system level effect compared to the *HP Turbine Rotor*, a complex component, due to its overarching effect on the overall system architecture.

3. VALIDATION OF STRUCTURAL COMPLEXITY METRIC

In order to establish the proposed metric as a valid measure of structural complexity, a series of both empirical and experimental validations is necessary. A positive outcome from a detailed validation procedure gives credence to the proposed metric and positions the metric for dissemination and application in engineered complex system development efforts. The first obstacle for validation is the inability to directly

measure complexity. Therefore we have to depend on indirect measures or well-accepted manifestation of complexity in terms of other system observables. The most visible of these system level observables is the system development cost.

We posit that the system development cost correlates super-linearly with Structural Complexity, as shown in fig. 6, and present two examples of empirical evidence from the literature [Wertz and Larson 1996, Wood et al. 2001, DARPA Report 2011]. They represent simple systems (e.g., hair-dryer family) at one end and highly complex satellite systems at the other end.
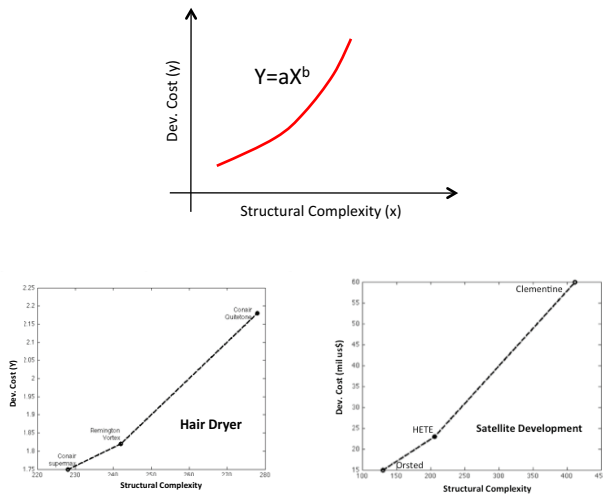

Figure 6. Predicted super-linear growth in development cost with increasing structural complexity and some empirical evidence from the literature [Wertz and Larson 1996, Wood *et al*. 2001, DARPA report 2011].

In both cases shown in fig. 6 above, if we were to fit a power law as per our hypothesis [Garvey 2000], we obtain R^2 = 0.99 and b = 1.16 and 1.3 respectively. But please bear in mind that these numbers are based on just 3 data points and therefore not statistically significant. They only support the trend consistent with our hypothesis but are not enough for a statistically significant confirmation.

Apart from such empirical evidence, we concentrated on conducting experiments with human subjects to see if we observe a similar behavior. These experiments were conducted as "natural" experiments as nearly as possible with a group of nearly homogeneous subjects, using simple ball and stick models (see fig. 7). They were asked to assemble molecular structures based on 2D pictures. Their assembly times, including any rework due to mistakes, were recorded. The sequence in which different subjects were given the molecular structures was randomized. In all cases, we assumed $\alpha = 0$ for all atoms, $\beta = 1$ for all links and $\gamma = 1/n$ where *n* is the number of atoms in a given molecule. This is because all atoms are used as is and there is no perceptible difference in assembling using different bond types (i.e., curved vs. straight bonds). Note that, in this particular experiment, sources of structural complexities are embedded within the connectivity structure and at the interfaces and not at the level of components (which are supplied as is). We are interested at validating the functional relationship between structural complexity and the development effort and not how individual components of structural complexity metric contribute to this relationship.

Initial results of our experimental investigations shows a similar relationship between structural complexity and molecule assembly time (see fig. 7) as that shown in fig. 6. Also note that variation in assembly time increases as the structural complexity level increases, validating our initial hypothesis. The most interesting preliminary result is the exponent of the power law relation, *b* = 1.51. This suggests that effort increases super-linearly but is not quite quadratic with increasing structural complexity. It would be instructive to investigate if parameter *b* increases with significantly higher structural complexity levels/regimes. This study will be expanded to include a larger sample size (i.e., number of test subjects).
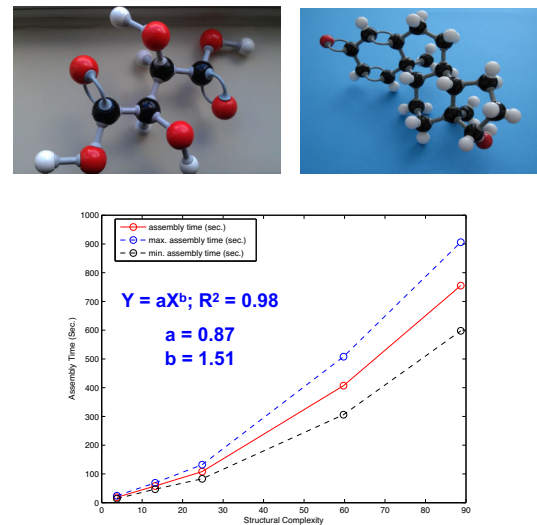

Figure 7. Some representative molecular structures that subjects were asked to assemble using a molecular kit and initial results based on a set of 5 structures with varying structural complexities. Here, number of subjects, N = 8.

Preliminary studies suggest that the parameters {a, b} are related to the organizational efficiency of the decision unit (i.e, the design and development team) or an actor, and the *complicatedness function* of the decision unit/actor respectively. The *complicatedness function* is the decision unit's perception of the associated complexity. It depends on other factors including modularity, cognitive bandwidth of the decision unit and its actors, system novelty, in addition to system complexity. The *complicatedness function* for human decision units / actor is an intermediate conduit through which structural complexity is manifested in terms of system development cost, which is a system observable. Modularity or design encapsulation is an effective way of good system organization and may reduce the system complicatedness.

6

Design encapsulation (notice that it also leads to information hiding) helps focus attention on a subset of the system at a time. This is similar to ''chunking'' of information to circumvent the human cognitive span limitation [Hirschi N.W, and Frey D.D., 2002]. A well-architected system may hide inherent complexity in an effective fashion such that it *appears* less complicated. This is the same as the idea of design encapsulation in the case of software systems. For example, a well-organized GUI (graphical user interface) might help reduce complicatedness of the system from a user's perspective, regardless of the underlying true system complexity. Inherent system complexity can be well organized and hidden from the observer, making the system appear less complicated.

4. IMPACT ON SYSTEM ARCHITECTURE

In this section, we look at the factors affecting distribution of complexity and structural modularity. It is shown that modularity and complexity are not necessarily negatively correlated, and we can have architectural configurations where both complexity and modularity increase at the same time. Let us explore the distribution of structural complexity among the system components using this simple model and investigate its implications on system development effort.

4.1 TOPOLOGICAL COMPLEXITY AND MODULARITY

Let us consider the following example of a system structure consisting of two modules or subsystems as shown below.
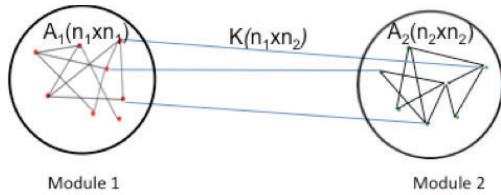

Figure 8. A hypothetical system divided into two modules or subsystems A_1 and A_2 and inter-module links characterized by rectangular matrix K.

The above system can be represented using a partitioned version of the adjacency matrix A and the associated topological complexity has an additional inter-module *integrative topological complexity* term, in addition to the intra-module topological complexities as shown below:

$$A = \left[\begin{array}{c|c} A_1 & K_2 \\ \hline K_1 & A_2 \end{array} \right] \Rightarrow E(A) = E(A_1) + E(A_2) + \Delta$$

Using the defined nomenclature, we arrive at the following expression for structural complexity of the system:

$$C = \left(\sum_{i=1}^{n_1} \alpha_i^{A_1} + \sum_{i=1}^{n_2} \alpha_i^{A_2} \right) + \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \beta_{i,j}^{A_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_2} \beta_{i,j}^{A_2} + \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_{i,j}^{K_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_1} \beta_{i,j}^{K_2} \right) \gamma E(A)$$

$$= \left(\sum_{i=1}^{n_1} \alpha_i^{A_1} + \sum_{i=1}^{n_2} \alpha_i^{A_2} \right)$$

$$+ \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \beta_{i,j}^{A_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_2} \beta_{i,j}^{A_2} + \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_{i,j}^{K_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_1} \beta_{i,j}^{K_2} \right) \gamma \left(E(A_1) + E(A_2) + \Delta \right)$$

$$= \underbrace{\left[\sum_{i=1}^{n_1} \alpha_i^{A_1} + \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \beta_{i,j}^{A_1} \right) \gamma E(A_1) \right]}_{module\ 1} + \underbrace{\left[\sum_{i=1}^{n_2} \alpha_i^{A_2} + \left(\sum_{i=1}^{n_2}\sum_{j=1}^{n_2} \beta_{i,j}^{A_2} \right) \gamma E(A_2) \right]}_{module\ 2}$$

$$+ \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \beta_{i,j}^{A_1} \right) \gamma E(A_2) + \left(\sum_{i=1}^{n_2}\sum_{j=1}^{n_2} \beta_{i,j}^{A_2} \right) \gamma E(A_1) + \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_{i,j}^{K_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_1} \beta_{i,j}^{K_2} \right) \gamma E(A_1)$$

$$+ \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_{i,j}^{K_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_1} \beta_{i,j}^{K_2} \right) \gamma E(A_2)$$

$$+ \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \beta_{i,j}^{A_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_2} \beta_{i,j}^{A_2} + \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_{i,j}^{K_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_1} \beta_{i,j}^{K_2} \right) \gamma \Delta$$

From the above expression, we can write, *Structural Complexity = sum of module structural complexities + integrative structural complexity.*

Let us apply the above result to a modular system structure with two modules in a slightly different way. We will define the intra-module topological complexities and also the integrative topological complexity as fraction of total topological complexity using three parameters $\{x_1, x_2, x_3\}$. Similarly express the intra-module and inter-module interactions as fraction of total interaction count using four parameters $\{y_1, y_2, y_3, y_4\}$. The detailed expressions are shown below.

$$E(A_1) = x_1 E(A); \ E(A_2) = x_2 E(A); \ \Delta = x_3 E(A)$$
$$Now, \ E(A_1) + E(A_2) + \Delta = E(A)$$
$$\Rightarrow x_1 + x_2 + x_3 = 1$$
$$m = \#interactions;$$
$$\sum_{i}^{n_1}\sum_{j}^{n_1} \beta_{i,j}^{A_1} = y_1 m; \ \sum_{i}^{n_2}\sum_{j}^{n_2} \beta_{i,j}^{A_2} = y_2 m; \ \sum_{i}^{n_1}\sum_{j}^{n_2} \beta_{i,j}^{K_1} = y_3 m; \ \sum_{i}^{n_2}\sum_{j}^{n_1} \beta_{i,j}^{K_2} = y_4 m$$
Assuming all interactions to be unity,
$$m = \sum_{i}^{n_1}\sum_{j}^{n_1} \beta_{i,j}^{A_1} + \sum_{i}^{n_2}\sum_{j}^{n_2} \beta_{i,j}^{A_2} + \sum_{i}^{n_1}\sum_{j}^{n_2} \beta_{i,j}^{K_1} + \sum_{i}^{n_2}\sum_{j}^{n_1} \beta_{i,j}^{K_2}$$
$$\Rightarrow y_1 + y_2 + y_3 + y_4 = 1$$
The aggregated component complexity is,
$$\Omega = \sum_{i=1}^{n_1} \alpha_i^{A_1} + \sum_{i=1}^{n_2} \alpha_i^{A_2}$$

Using the earlier derivation of structural complexity for two sub-system example, we can write the structural complexity in this case as:

Copyright © 2013 by ASME

$$C = \Omega + \gamma m E(A)\underbrace{[x_1 y_1 + x_2 y_2 + x_1 y_2 + x_2 y_1 + x_3 + (x_1 + x_2)(y_3 + y_4)]}_{=1}$$

Hence, there are seven parameters and only two constraints: $x_1 + x_2 + x_3 = 1$ and $y_1 + y_2 + y_3 + y_4 = 1$. They shape how the overall complexity is distributed within and across the modules and the associated structural modularity. Modularity index, Q is defined as the fraction of edges that fall within module 1 or 2, minus the expected number of edges within module 1 and 2 for a *random graph* with same node degree distribution as the given graph. Expanding this basic definition and after unpacking different components, we arrive at the final form of the modularity index for this two module system as:

$$Q = \sum_{c=1}^{2} (e_{ii} - a_i^2)$$

where e_{ii} is the fraction of edges with both end vertices in the same module i and a_i is the fraction of edges with at least one end vertex inside module i. In this case, given the modules a priori and using the two constraint relations on the seven parameters, we arrive at the modularity index:

$$Q = (y_1 + y_2) - \left[(y_1 + y_3)^2 + (y_2 + y_4)^2 \right]$$
$$= (y_1 + y_2) + 2(y_2 + y_4) - 2(y_2 + y_4)^2 - 1$$

There is conventional wisdom that complexity and modularity are negatively correlated and increased modularity brings down the structural complexity [Baldwin and Clark, 2000; Lindemann et al., 2008]. This is not a causal relation and what we have is a set of seven fractions - $\{x_1, x_2, x_3\}$ and $\{y_1, y_2, y_3, y_4\}$ that determine how the overall topological complexity and modularity are related. This argument also holds for structural complexity, which is in a sense, an affine transformation of topological complexity for given component and interface complexities. We can very well have increases in complexity alongside an increase in modularity as shown below (see fig. 9).
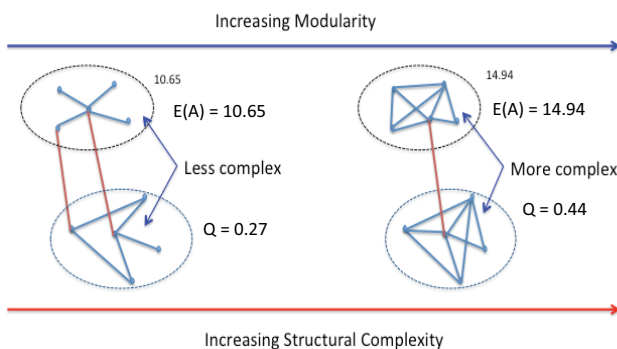

Figure 9. Increasing modularity does not necessarily reduce topological complexity. For a 2 module example here, it depends on 7 parameters that

shape structural complexity vs. modularity space. Newman's modularity metric Q was used to compute modularity.

The seven parameters described above shape how the overall complexity is distributed within and across the modules and the associated degree of modularity.

4.2 DISTRIBUTION OF STRUCTURAL COMPLEXITY AND ITS IMPLICATIONS ON SYSTEM DEVELOPMENT EFFORTS

Distribution of structural complexity across the system elements play a very significant role in achieving a set of system properties and often to programmatic success of the system development project. Knowledge of overall system architecture is absolutely critical to be able to quantify and track the complexity during the system development activity. There may be subsystems that are significantly more complex and respective development teams should be able to handle such high complexity in order to be successful.

We might view the system from a more abstract perspective / viewpoint where the modules or subsystems are treated as super-components. Each super-component has an internal complexity (that represents the complexity of the super-component) and this fact should not be overlooked. The total structural complexity is now distributed within and across the subsystems. This aspect can be best explained with a simple example as shown in fig. 10 below.
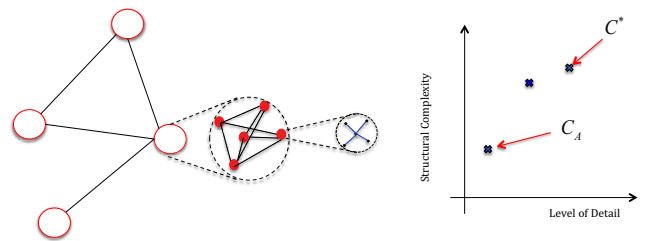

Figure 10. Increasingly detailed view of the system and evolution of structural complexity if we assume components at each level to be of similar category.

At top level, each component actually represents a subsystem or module and their lower level details are shown in fig. 10. Complexity estimates are performed at each level of detail without considering the fact that at the top level, we do not have component, but we have subsystems and we cannot treat them as simple components. Doing that one might get the impression that structural complexity is only C_A while in reality, it is C^*. This leads to a gross underestimation of structural complexity of the system.

In order to extract information on complexity distribution we do need complete information about the internal structure of subsystems. This information is crucial for tracking and management of large, engineered system development efforts. Implication of the complexity distribution on system development effort and associated decision-making can be best explored using case studies and development of *Boeing 787* (e.g., the *Dreamliner*) is a good example [Heimsch 2011, Cohan 2011]. The Boeing Company announced in 2004 that it

Copyright © 2013 by ASME

was embarking on an ambitious commercial airplane development project in order to bring the 787 Dreamliner to market. Recognizing the need for speed to market for the 787, along with increased quality standards and reduced production costs, Boeing focused on an innovation strategy and decided to outsource 70% of the design and manufacturing for this plane [Heimsch 2011, Cohan 2011]. Boeing felt that the 787 Dreamliner represented groundbreaking innovation, with benefits that would resonate with customers and enable Boeing to regain its leadership position and the 787's composite material design, aerodynamics, fuel efficiency and propulsion systems has redefined how commercial aircraft are designed and manufactured, and will impact the broader aviation industry [Heimsch 2011, Cohan 2008]. Considered the largest industrial program in the world, Boeing chose to partner with 17 companies in 10 different countries and the outsourcing of the design and production amounts to 70% of the aircraft. Historically, Boeing has outsourced much of its manufacturing, considering it non-core to its operations. But Boeing has focused on owning its core design work, viewing it as its competitive advantage. However, with the 787, Boeing opted to also outsource much of the development work. Arguably, the decision to outsource so much of the design and production of the 787 might have played a significant role in the project not meeting its ambitious goals in terms of reduced production cost and speed to market. The development project posed a big management challenge and led to a three-year delay in launching the first aircraft for operation and resulted in significant financial loss for Boeing. By outsourcing both the design and the manufacturing, Boeing temporarily lost control of the development process. This is to say that Boeing had a clear view only at the level of primary modules, but not beyond. This view obscures what is inside the subsystems and made it difficult for Boeing to judge the total structural complexity of the system as it evolved. If a subsystem or module started to become too complex, it is possible that the outsourcing partner did not have the adequate capability for handling that level of complexity and this may have jeopardized the overall system development effort [Cohan 2011]. At the top level, there is a tendency to assign a lower complexity to subsystems as all details may not be available early on. This inability to track and manage such complexity growth actively at both subsystem and system level, may lead to suboptimal results and sometimes programmatic failure. In case of evolving system architectures, one should keep track of the overall evolving architecture to make sure subsystem complexities are contained within sustainable limits. If not, there might be a need to re-structure the subsystem development team to address the evolving reality of asymmetry in complexity distribution among various subsystems. It is imperative that every large-scale system development effort does active complexity distribution and management.

In this light, we argue that the importance of complete knowledge of the overall system architecture is crucial for decision-making during the development process and constitutes a core capability for the primary system development organization. This capability is essential for complexity to be tracked and actively managed during the process. It is argued that delivering the system properties is ultimately the responsibility of primary system development organization with an appropriate integrative mechanism where the overall system architecture remains visible.

5. DISCUSSIONS

In this work, we formulate the structural complexity metric for engineered complex systems, which is shown to consist of three terms representing complexities of system components, connections among these components and topological complexity. We introduced the notion of *matrix energy* as a measure of topological complexity of product architecture as a rigorous measure of topological complexity. Topological complexity metric also shares properties similar to and found to correlate strongly with information-theoretic complexity metrics for networks. Topological complexity increases from *centralized* towards more *distributed* architectures. We have performed mathematical / conceptual verification of the metric using Weyuker's criteria. Operationalization of this metric in the context of product architecture is demonstrated.

Sensitivity analysis due to system component removal was presented and its application to an aircraft engines hinted at the systemic importance of secondary system over the primary functionality generators as the architected systems become more distributed in nature.

In order to establish the proposed metric as a valid measure of structural complexity, we presented empirical evidence from literature and experimental validations are being performed to further buttress the validation aspects of the proposed complexity metric. We have posited a super-linear growth in development cost with increasing structural complexity and presented confirmatory evidence from existing literature and preliminary experimental validation. We also hinted at existence of *complicatedness function* for human decision units/actor through which structural complexity is manifested in terms of system development cost, which is a system observable.

We demonstrated that structural complexity and modularity are not necessarily negatively correlated, against a popular conventional wisdom. We further discussed distribution of complexity across the system architecture and argued that adequate knowledge and visibility of the overall system architecture is absolutely essential for matured complexity management capability. This leads to eventual success of the system development activity. Distribution of overall complexity is a critically important and has a big impact on system architecting. If there are significantly more complex subsystems, the development team should have the capability to handle this high complexity. Knowledge of the relative subsystem complexities influences the selection / composition of the subsystem development team. It is imperative for every large-scale system development effort to have active complexity distribution and management capability.

Copyright © 2013 by ASME

Going forward, the proposed structural complexity metric can serve future complexity-based product design and optimization framework and help explore important questions related to tracking, management and distribution of structural complexity across the system architecture and its impact on other system performance/lifecycle measures.

ACKNOWLEDGMENTS

REFERENCES

Denman J., Sinha K., de Weck O., "Technology Insertion in Turbofan Engine and assessment of Architectural Complexity", *13th International DSM Conference*, Cambridge, MA, September 14-15, 2011.

Baldwin, C.Y and Clark, K.B, "Design Rules", MIT Press, 2000.

Fabrikant A., Koutsoupias E., Papadimitriou C., "Heuristically Optimized Trade-Offs: A New Paradigm for Power Laws in the Internet", *Automata, Languages and Programming*, Vol. 2380, January 2002.

Horn and Johnson, "Topics in Matrix Analysis", Cambridge Press, 1994.

Hirschi N.W, and Frey D.D., "Cognition and complexity: An experiment on the effect of coupling in parameter design", Research in Engineering Design 13, 2002, pp 123 – 131.

Ivan Gutman, Tanja Soldatovic and Dusica Vidovic, "The energy of a graph and its size dependence. A Monte Carlo approach", Chemical Physics Letters 297 (1998), 428–432.

Agte J., Borer N. and de Weck O., "Design of Long-Endurance Systems With Inherent Robustness to Partial Failures During Operations", *J. Mech. Des.* 134(10), 2012.

Lindemann U, Maurer M, and Braun T., "Structural Complexity Management - An Approach for the Field of Product Design" – Springer, 2008.

Liu, D., Wang, H., and Van Mieghem, P., "Spectral Perturbation and reconstructability of complex networks", Physical Review E 81, 016101, 2010.

Peter S. Cohan, "You Can't Order Change: Lessons from Jim McNerney's Turnaround at Boeing", Portfolio Hardcover, ISBN-10: 1591842395, 2008.

Piet Van Mieghem (2011): Graph Spectra for Complex Networks, Cambridge University Press, ISBN 978-0-521-19458-7, 2011.

Sheard S.A. and Mostashari A., "A Complexity Typology for Systems Engineering", Systems Engineering, 2010.

Sinha K., de Weck O., "Structural Complexity Metric for Engineered Complex Systems and its Application", *14th International DSM Conference*, 2012.

Tang.V, Salminen.V : "Towards a Theory of Complicatedness: Framework for Complex Systems Analysis and Design" , 13th International Conference on Engineering Design, Glasgow, Scotland, August 2001

Ulrich, K.; Eppinger, S. (1995): Product Design and Development. New York: McGraw-Hill 1995.

V. Nikiforov, "The energy of graphs and matrices", J. Math. Anal. Appl. 326 (2007), 1472–1475.

Weyuker E. "Evaluating software complexity measures", IEEE Transactions on Software Engineering, 1988, 14 (9), 1357-1365.

Wood *et al*., "Interfaces and Product Architecture", ASME DETC, 2001.

Wertz J.R. and Larson W.J., "Reducing Space Mission Cost", Springer (1996), ISBN-10: 1881883051.

Garvey P.R., "Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective", CRC Press (2000), ISBN-10: 0824789660.

Sadin, Stanley R., Povinelli, Frederick P.; Rosen, Robert (1988), "The NASA technology push towards future space mission systems", 39th International Astronautical Congress,1988.

Wilhelm, T.; Hollunder, J., "Information theoretic description of networks", Physica A 2007, 385, 385–396.

http://www.darpa.mil/Our_Work/TTO/Programs/AVM/AVM_Design_Tools_(META).aspx, 2011

Dave Heimsch, "http://www.scribd.com/doc/59779160/Boeing-Dream-Liner-Outsourcing-Study", 2011.

Cohan P. S. ,"http://www.dailyfinance.com/2011/01/21/boeing-dreamliner-delays-outsourcing-goes-too-far/", 2011.