Concurrent Engineering

http://cer.sagepub.com

Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints

Katja Hölttä-Otto and Olivier de Weck

Concurrent Engineering 2007; 15; 113
DOI: 10.1177/1063293X07078931

The online version of this article can be found at:
http://cer.sagepub.com/cgi/content/abstract/15/2/113

Published by:
SAGE Publications
http://www.sagepublications.com

Additional services and information for *Concurrent Engineering* can be found at:

Email Alerts: http://cer.sagepub.com/cgi/alerts

Subscriptions: http://cer.sagepub.com/subscriptions

Reprints: http://www.sagepub.com/journalsReprints.nav

Permissions: http://www.sagepub.com/journalsPermissions.nav

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

CONCURRENT ENGINEERING: Research and Applications

Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints

Katja Hölttä-Otto[1],* **and Olivier de Weck**[2]

[1]*Department of Mechanical Engineering, University of Massachusetts*
Dartmouth 285 Old Westport Rd, North Dartmouth, MA 02747-2300, USA
[2]*Department of Aeronautics and Astronautics and Engineering Systems Division*
Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139

Abstract: There is consensus that modularity has many benefits from cost savings due to increased commonality to enabling a higher variety of products. Full modularity is, however, not always achievable. How engineering systems and products whose design is heavily influenced by technical constraints, such as weight or size limitations, tend to exhibit rather integral architectures is shown in this study. For this, two metrics are defined on the basis of a binary design structure matrix (DSM) representation of a system or product. The non-zero fraction (NZF) captures the sparsity of the interrelationships between components between zero and one, while the singular value modularity index (SMI) captures the degree of internal coupling, also between zero and one. These metrics are first developed using idealized canonical architectures and are then applied to two different product pairs that are functionally equivalent, but different in terms of technical constraints. Empirical evidence is presented that the lightweight variant of the same product tends to be more integral, presumably to achieve higher mass efficiency. These observations are strengthened by comparing the results to another, previously published, modularity metric as well as by comparing sparsity and modularity of a set of 15 products against a control population of randomly generated architectures of equivalent size and density. The results suggest that, indeed, some products are inherently less modular than others due to technological factors. The main advantage of SMI is that it enables analysis of the degree of modularity of any architecture independent of subjective module choices.

Key Words: complexity, modularity, system engineering.

1. Introduction

Modularity, modular product architectures, and modular product platforms have recently gained popularity in product development. Modular architectures, consisting of loosely coupled chunks (modules), have many benefits from cost savings due to commonality to independent design of modules. However, a fully modular design may not always be achievable when designing complex systems and products. It is suggested that modularity is not simply a binary characteristic, but that products can and do exhibit *varying degrees of modularity*. In this study, the authors first develop a new measure of modularity and then show that designs that are heavily driven by weight, size, or other performance constraints, often exhibit rather integral architectures commensurate with a smaller degree of modularity. In contrast, when the design is driven mainly by business goals, such as cost savings or commonality, a more modular architecture often emerges. This is shown by quantitatively analyzing the degree of modularity of existing product architectures and qualitatively discussing the relationship between technical and business constraints. A set of two product pairs (telephones, computers) is used that represent business and technical constraint driven versions of the same type of product to illustrate the findings.

2. Motivation

A module is commonly defined as an independent chunk that is *highly coupled within, but only loosely coupled to the rest of the system*. As such, knowing the elements of form of a system and the interconnections between them should be enough to assess modularity. Additionally, modular architecture is sometimes defined as having a one-to-one mapping from functional elements in the function structure to the physical components of the product [1]. This is also supported by the independence axiom in axiomatic design [2]. The axiom states that functional requirements should be kept uncoupled with the design parameters.

*Author to whom correspondence should be addressed.
E-mail: katja.holtta-otto@umassd.edu

1063-293X/07/02 0113–13 $10.00/0 DOI: 10.1177/1063293X07078931

© 2007 SAGE Publications
Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

The uncoupled design can lead to modules that are loosely connected to one another as the connections are often due to the functions of the modules. The loose connections result in many potential advantages: ease of decomposing and assigning design tasks to various groups, encapsulating functions in modules that are easy to upgrade and change in the future, increased robustness, and so forth.

Modularity has two fundamental, although possibly related, aspects that are discussed separately:

(1) *Coupling*: Systems where the strength of coupling within certain chunks of form is stronger than the strength of coupling on average across the system are defined to be 'modular.' Conversely, systems that have a high density of interconnections throughout (every component connects to nearly every other component) or that feature a few highly connected components that connect to almost every other part of the system are referred to as 'integral'.
(2) *Functional Encapsulation*: Systems where functions are clearly executed by a subset of components and where this subset of components is highly coupled, as described above, are modular. On the other hand, systems where one component carries out many functions or where individual functions are assigned to many components (that do not necessarily cluster) are referred to as integral architectures.

In this study, the form–form modularity of a system is quantitatively assessed based on coupling (1) and only make qualitative statements about implications for function–form modularity (2).

For a number of systems axiomatic design has been shown to be an effective approach to system design [2]. Examples of machines designed axiomatically to be modular or decoupled include machine tools, wafer polishing machines, and other industrial equipment. One may wonder why modularity has then not made greater inroads in other types of products such as automobiles, aircraft, or portable electronic devices. *Is there an inherent tradeoff between modularity and some technical constraints such as lightweighting*? It has been speculated that modular design is less efficient in terms of packing density, mass, and other technical aspects. This is the main point sought to be addressed in this study.

One answer could be that uncoupled or decoupled design often leads to a modular product architecture that features many interfaces with potential interface losses and a suboptimal use of space, mass, and energy during operations. Indeed, generally, there is a price to be paid for modularity in terms of other product characteristics. For example, Whitney argues that modular architectures tend to have more parts, tend to be uncoupled or decoupled, and favor 'business performance,' whereas integral architectures have fewer parts, tend to be coupled, and favor 'technical performance' [3]. Some high performance systems, such as automotive and aerospace vehicles appear to favor highly coupled architectures, where one part fulfills potentially many functions. An example is the new blended wing body (BWB) concept developed by Boeing [4]. In this radical design the differences between wing, fuselage, and empennage are blurred, i.e., blended. This blending leads to significantly higher efficiencies in terms of mass utilization, lift generation, and ultimately a lower projected fuel burn per passenger per mile flown, but also reduces the degree of modularity of the system, relative to the traditional tube-and-wing aircraft architecture.

In summary, there appears to be a potential trade-off between the desire for modularity from a 'business' standpoint and the desire for high performance and efficiency in the technical domain. This leads to the following hypothesis that, if proven correct, would be an important principle for system and product architecting:

Hypothesis (1)

A high degree of coupling among elements of form can be the result of imposing stringent technical constraints – such as lightweighting, tight packaging, power efficiency, or latency minimization – during complex system design.

3. Related work

A. Modularity

Modularity brings both advantages and disadvantages. Modularity often means using the same module in multiple products enabling a large variety of products while using more common component types than if the different products did not share common modules. This brings scale and scope advantages, such as reduced capital requirements, and economies in parts sourcing and manufacturing [5–7].

Modules are also helpful in design re-use since already designed modules with well-defined interfaces can be used again in other designs. This applies to software products as well as hardware. Design re-use can lead to reduced cycle time, which in turn results in for example increased revenue due to increased market penetration as a result of being first to market, success in time sensitive markets, and shorter time to market increases accuracy of meeting emerging customer needs.

Product change, upgrade, and variety can potentially be achieved by replacing one or more modules in a system without other changes to the overall product, or product platform. This is also useful in delaying differentiation in made-to-order products [8].

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

In addition, a well-defined module, in terms of simple interfaces, can ease project management due to decoupling of tasks and providing design freedom within a module. Modularity also makes a complex product architecture appear simpler and therefore easier to manage, i.e., it reduces apparent complexity [1,5]. Further, modularity can also bring benefits at the end of a product's lifecycle – as a means to ease the disassembly and recycling of the product [9]. Modularity can be considered during assembly, but here the authors are most interested in modularity as it applies to finished products in their operational state.

The above advantages of modularity are mainly related to the 'business performance' of a manufacturing firm as opposed to the 'technical performance' of the product itself.

Owing to the many advantages of modularity, many methods to define modules have been proposed [5,10–12]. Interestingly, Holtta and Salonen [13] showed that these methods give different suggestions for a modular architecture, even when starting with the same initial conditions. As an example, these methods disagree on the appropriate degree of modularity (or integrality) of the architecture. The modular function deployment method [5], for example, tends to lead to more integral architectures due to its limit on the number of allowed modules.

Kusiak and Huang [14] presents another algorithm to develop modular multi-chip modules. He integrates product and process design. His algorithm [14] includes a step to define the upper bound for module size (level of integrality) but no clear rule on how to determine the size. Another method, modularization by integration [15], includes a step toward managing the trade-off between modularity and integrality. This method takes functional and economic issues as well as product variety into account, but even though functionality is thought of, technical constraints, such as performance, are not included in the method's criteria. This is true in all methods because performance, weight, etc. are constraints, not design parameters and are therefore difficult to take into account when making product architecture decisions about modularity.

The question remains how to decide on the degree of modularity or on the number of modules. Ericsson and Erixon [5] argue that the ideal number of modules is approximately the square root of the number of parts. This is based on minimizing assembly time. Braha [16], on the other hand, suggests deciding on a team size based on the maximum number of attributes that a team can handle, and that this can be used to decide what to include into modules. These are two suggestions, but so far there appears to be no universally accepted method for defining the degree of modularity in a specific case.

B. Modularity and Integrality

A number of authors have addressed the differences of modular and integral architectures. Sosa et al. [17] introduce a method to identify whether a system is modular or integral based on the component interactions in the system design structure matrix (DSM) [18,19]. They also note the importance of identifying the integrality or the modularity of the system, because it highly impacts the design team interactions. Also, Ulrich [1] points out that the product development process is different for modular and integral products. Neither author, however, says when a modular system is more appropriate than an integral one. Rather, they state that the design process must be adjusted for modular and integral architectures. Along the same lines, Fixson and Clark [20] discuss the effects of the architecture type on the costs along the supply chain.

A few authors discuss the local versus global performance of a product architecture. Ulrich [1] discusses many advantages of modularity as well as technical constraints in the design. He says that modularity can help optimize local performance characteristics. For global performance he argues that integral architecture is better, especially if there are weight and size constraints. Ulrich discusses the topic thoroughly, but only qualitatively. Also Gonzalez-Zugasti and Otto [21] show that some performance is sacrificed to obtain goals of the individual products that are created from a common platform. They claim that the ideal would be to give up some small amount of performance locally in modules in order to achieve a better common platform.

Regardless of the many benefits of modularity, full modularity may not always be achievable. Whitney [22] argues this is especially in the case of high power mechanical products, as opposed to low power information processing products. A more modular product is likely to be larger, heavier, slower, and less energy efficient. Also side effects are harder to control. This argument is supported also by Cutherell [23]. Whitney [22] compares complex electromechanical-optical products to large chips designed with very-large-scale-integration (VSLI), which can be considered fully modular, and are in line with Suh's design axioms [2]. Mechanical parts have a 'multi-function character' partly due to basic physics (material contains also energy, rotating axle transmits shear loads and rotational energy) and partly due to 'design economy'. Whitney also points out that the interfaces (in high power systems) require substantial space and weight and they must be custom-designed for each application, thus making modular design more difficult. In addition, Benini and de Micheli [24] discuss the same issue. According to them power optimization is especially important in low power, high performance systems,

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

such as cellular phones. Cutherell [23] mentions, as an example, a heavier car being less fuel efficient. He also uses a hand-held electronic calculator as an example of an almost fully integral (excl. battery) architecture in order to make the product lightweight. Software design faces similar issues. Kazman et al. [25] and Bass et al. [26] discuss the architectural trade-offs in software design. They introduce a method to identify trade-off points between system modifiability, availability, and performance in order to achieve wanted system quality.

There is an abundance of literature on the benefits of both modularity and integrality but the evaluation of which is better and when is often subjective, qualitative, or speculative. In this study the authors begin development of a more quantitative approach for quantifying modularity-versus-integrality and what effect technical constraints may have on the degree of modularity of functionally equivalent products.

4. Approach

In this section, the authors describe the methods of analysis used during this research. The main method is to use DSM to analyze the form–form coupling and modularity of a system or product [18]. Dong and Whitney [27] showed that Suh's [2] design matrix (DM) and DSMs are closely linked and that most couplings in the one matrix are due to the couplings in the other matrix. This result allows the researchers to focus their effort on analyzing the structure of form–form DSMs and ignore for now any issues associated with function–form allocation. In other words, the form–form coupling is analyzed exclusively to quantify modularity, not functional encapsulation in elements of form.

The validity of hypothesis (1) can be probed by analyzing the degree of modularity of a range of products and systems and comparing this to the technical and business constraints of those same products. Two modularity metrics are used, one developed here, and one from the literature.

The latter modularity metric was developed by Guo and Gershenson [28]. The metric is based on the definition of a module, where a module is tightly connected within a module and loosely connected to the rest of the system. They define modularity as:

$$M_2 = \frac{\sum_{k=1}^{M}\left(\sum_{i=n_k}^{m_k}\sum_{j=n_k}^{m_k} R_{ij}/(m_k-n_k+1)^2\right) - \sum_{k=1}^{M}\left(\sum_{i=n_k}^{m_k}\left(\sum_{j=1}^{n_k-1} R_{ij} + \sum_{j=m_k+1}^{N} R_{ij}\right)/(m_k-n_k+1)(N-m_k+n_k-1)\right)}{M}, \tag{1}$$

where n_k is the index of the first component in the kth module; m_k is the index of the last component in the kth module; M is the total number of modules in the product; N is the total number of components in the product; and R_{ij} is the value of the ith row and jth column element in the 'modularity' matrix.

This metric can be applied to component-to-component (form–form) DSMs.

The authors first analyzed existing products using this modularity metric. This gave certain results, but it also required an *a priori* subjective definition as to where the module boundaries are or should be. The goal, however, is to find the fundamental degree of coupling of a product independent of where the module boundaries are set or how the rows and columns of a DSM are sorted. Thus the SMI was developed based on the decay pattern of the singular values of the binary DSM describing the interconnections between components. This metric is described in detail in Section V and then applied in Sections 6 and 7.

Further, in order to be able to compare quantitatively a technical constraint related to volume, a so-called *packaging factor* was calculated for a set of products. The packaging factor is defined as:

$$P = \frac{V_{\text{unused}}}{V_{\text{total}}} \tag{2}$$

where P is the packaging factor; V_{unused} is the volume of the unused empty space inside a product; and V_{total} is the total exterior volume of the product.

Note that any empty space with a purpose, such as space between fan blades or space for a battery, is not included in V_{unused}. The exterior volumes were determined by submerging the product in water in a vacuum bag, measuring the displaced water, and then subtracting the volume of the bag. The volume of the unused space was determined by carefully and completely filling the unused spaces with a soft crafts molding product, and determining the volume of the molding product.

The research approach consists of the following steps:

(1) Select a variety of engineering systems (products) in terms of business and technical constraints. This includes two pairs of functionally equivalent products: cellular and desk telephones as well as laptop and desktop computers.
(2) Decompose the elements of form of these selected systems.
(3) Map the interconnections between elements of form by applying DSM.

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

(4) Calculate modularity indices for each system (using M_2 and SMI).
(5) Calculate the packaging factors for each system.
(6) Discuss the commonalities and differences between the system architectures, particularly with respect to the degree of modularity and technical and business constraints.
(7) Compute modularity (and sparsity) for a larger set of functionally distinct products.
(8) Based on the previous observations, derive some generalizable statements about the trade-off between the degree of modularity and system performance.
(9) Conclude by either supporting or refuting the hypothesis (1).
(10) In the following section the authors introduce the new metric and then move on to analyze the different systems and products in the remainder of the study.

5. Singular Value Modularity Index (SMI)

The metric introduced originally in an earlier work [29] and further developed here is an unambiguous way to quantify the degree of modularity of a product based on its internal connectivity structure. This structure can be represented by a DSM, where the diagonal entries are zeros and off-diagonal elements are set to a non-zero value (typically '1') if two components are connected. The direct connection between two components can be based on physical contact, material flow (e.g., a working fluid), energy flow, or information flow [10,18]. Using a non-binary DSM enables differentiation between connections of different strengths. However, the examples here use a binary DSM for the sake of simplicity.

Figure 1 shows idealized examples of product structure with $N=7$ components each. In the first case, every component connects to every other component. In the second case, one component connects to all other components, but none of the other components connect to each other, and in the third case components only connect to their direct neighbors.

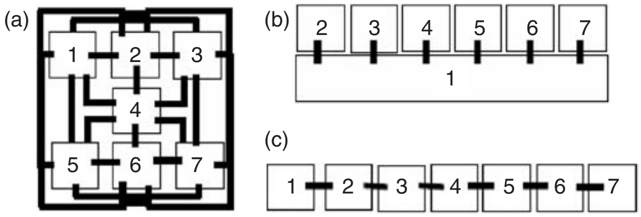


Figure 1. (a) Fully 'integral' system, (b) 'bus-modular' system, and (c) fully 'modular' system.

The binary matrices for these systems are given as:

$$\mathrm{DSM}_a = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$\mathrm{DSM}_b = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

$$\mathrm{DSM}_c = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

These matrices can also be visualized in terms of their sparsity pattern (Figure 2).

Performing a singular value decomposition (SVD) on the binary DSM matrix reveals its singular values (the singular values are the square roots of the eigenvalues of $\mathrm{DSM}^{\mathrm{T}}\mathrm{DSM}$) and corresponding orthogonal eigenvectors:

$$\mathrm{DSM} = U \cdot \sum\nolimits_{\mathrm{DSM}} \cdot V^{\mathrm{T}}, \tag{4}$$

where

$$\sum\nolimits_{\mathrm{DSM}} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_N \end{bmatrix}. \tag{5}$$

The singular values, $\sigma_1, \sigma_2, \ldots, \sigma_N$ are naturally sorted in descending order and N is the number of components

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

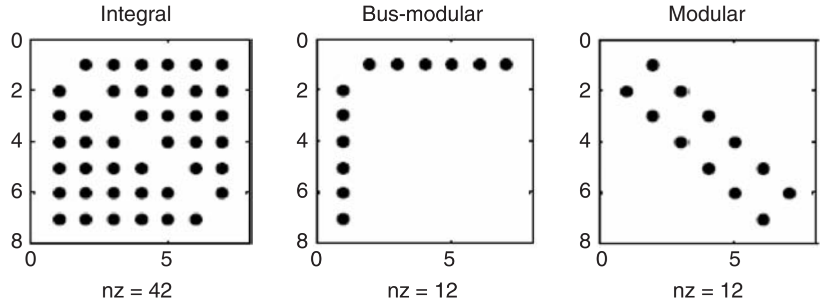


Figure 2. Sparsity pattern for idealized DSMs: (a) fully integral, (b) bus-modular, and (c) modular.

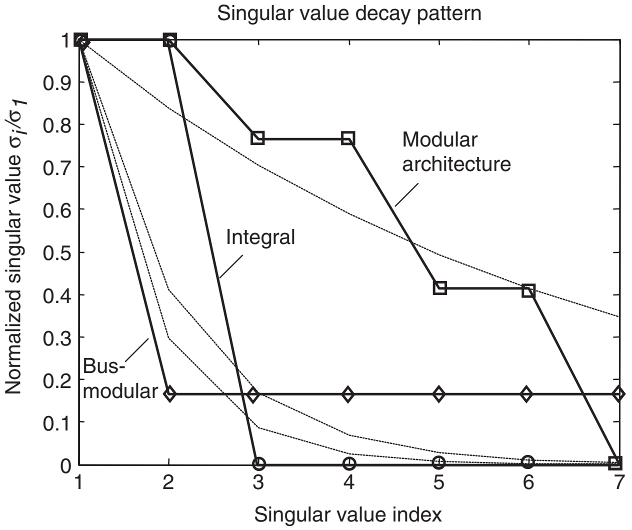


Figure 3. Singular values of the 'integral', 'bus-modular,' and 'modular' systems shown in Figure 1. Exponential approximation of decay patterns according to Equation (6) are shown by dashed lines.

(= number of rows and columns in the DSM) of the system. Figure 3 contains a plot of the normalized singular values for the three systems (Figure 2(a)–(c)) shown in decreasing order of magnitude, respectively. Normalization is accomplished by dividing by σ_1, the largest singular value.

There is a significant difference in the singular value decay structure among the systems. The integral system has one large normalized singular value (1.0) followed by $N-1$ smaller ones (at a value of 1/6–0.17), the bus-modular system has two large non-zero singular values (1.0) and the remaining singular values are all zero. The fully modular system shows a much more gradual decay of its singular values. This can be explained by the way in which the original DSM can be reconstructed completely with the (normal) eigenvectors and their corresponding non-zero singular values, see Equation (4). In other words, the information that describes the system is more broadly distributed in modular versus integral architectures.

Moreover, singular values are often used for balanced model reduction in linear systems theory [30]. A system with a high decay rate of singular values can more easily be reduced to a smaller set without much loss of information. Similarly, an integral system can be described well by focusing only on the most connected components. A modular design, on the other hand, requires more information to be described completely, relative to a bus-modular or integral system. In other words, all singular values except one are non-zero (Figure 3) and $N-1$ eigenvectors must be retained for a complete description of the modular system.

A modularity index is therefore postulated that reflects the degree to which the important information for describing system connectivity is concentrated in a few components that are highly connected across the system. Such integral systems show a much more rapid drop-off in the magnitude of their singular values, relative to modular systems where such information is more widely distributed throughout the system. The SMI therefore measures the decay rate of the sorted, normalized singular values in the system:

$$\mathrm{SMI} = \frac{1}{N} \arg\min_{\alpha} \sum_{i=1}^{N} \left| \frac{\sigma_i}{\sigma_1} - e^{-[i-1]/\alpha} \right|. \qquad (6)$$

In Equation (6), it is assumed that singular values in all systems decay exponentially according to $\exp(-(i-1)/\alpha)$, but that they decay more slowly in modular versus integral systems. The SMI is then equal to the quantity α^*/N that minimizes the error between an exponential decay and the actual decay structure across all singular values. The larger the SMI, the more modular is the system. The virtues of SMI are as follows:

(1) This index is theoretically bounded between 0 and 1.
(2) An SMI closer to 1.0 indicates a higher degree of modularity, where the connectivity information is more broadly distributed throughout the system. An SMI closer to 0 indicates a more integral system.
(3) The SMI is independent of subjectively drawn module boundaries or how the rows and columns of the underlying binary DSM are sorted.
(4) The SMI is scale-free, i.e., it can be computed for systems of different sizes that have the same

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

Table 1. Values of SMI and NZF for idealized architectures with N = 7.

Case	Description	N	α	SMI (Equation (6))	NZF (Equation (7))
(a)	Integral	7	1.12	0.16	1.0
(b)	Bus-modular	7	0.82	0.12	0.29
(c)	Modular	7	5.67	0.81	0.29

fundamental architecture and it will return the same (or nearly the same) value.

In addition to the SMI the non-zero fraction (NZF) is also computed, which is the fraction of non-zero entries in the DSM, after the main diagonal has been removed. This is computed as:

$$\mathrm{NZF} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{DSM}_{ij}}{N(N-1)}. \qquad (7)$$

The NZF gives an indication of sparsity of the underlying DSM. A value of $\mathrm{NZF} = 1$ means that all components are connected to all other components. A low value of NZF, say $\mathrm{NZF} < 0.15$ indicates that the system is generally sparsely connected relative to its size. NZF and SMI are both computed on a scale from 0 to 1, since they are normalized metrics, but they measure different things. NZF measures the sparsity of a system, while SMI measures its degree of modularity. Both metrics will be applied to the analysis of existing products and systems in Sections 6 and 7. The SMI and NZF for the idealized cases (Figure 1) are shown in Table 1.

The modular architecture has the largest value of SMI, as expected. The value of SMI is not exactly 1.0 due to the assumption of exponential decay. Even though the bus-modular and modular architecture are both sparse, the bus-modular architecture has a significantly lower value of SMI, indicating that it is in fact much closer to the integral architecture in terms of connectivity. A change to the single highly connected component has the potential for affecting all other components in the system. In the next section these metrics are applied to pairs of functionally equivalent products that are subject to different technical constraints.

6. Modularity of Functionally Equivalent Products

First, a set of products is selected in terms of their dominant technical and business constraints. Products are chosen in pairs so that they share similar functionality but differ in terms of dominant business and technical constraints. Two pairs of products are included in the analysis: a cellular and desk telephone as well as a laptop and desktop computer. Then the elements of form of these selected systems are decomposed and component-to-component DSMs are generated for each system.

The first pair of products (Pair A) is a cellular phone and a desk phone. In order for the functionality to be as similar as possible between the two phones, a basic cellular phone is chosen that did not include extra functionality compared to the desk phone. The cellular phone represents a product where technical constraints (e.g., weight and power consumption) are the determining factor in design, whereas there are no such constraints involved in the design of the desk phone, or if they are present the constraints are substantially relaxed.

The two products are decomposed and the following DSMs are derived for them (Figure 4).

The second pair of products is a pair of computers. Here a laptop represents a product where design technical constraints influenced the architecture during design. The desktop PC, on the other hand, is designed primarily from a business performance point of view (e.g., flexible to change, separate development). The two computers are decomposed and the following DSMs are derived for them (Figure 5).

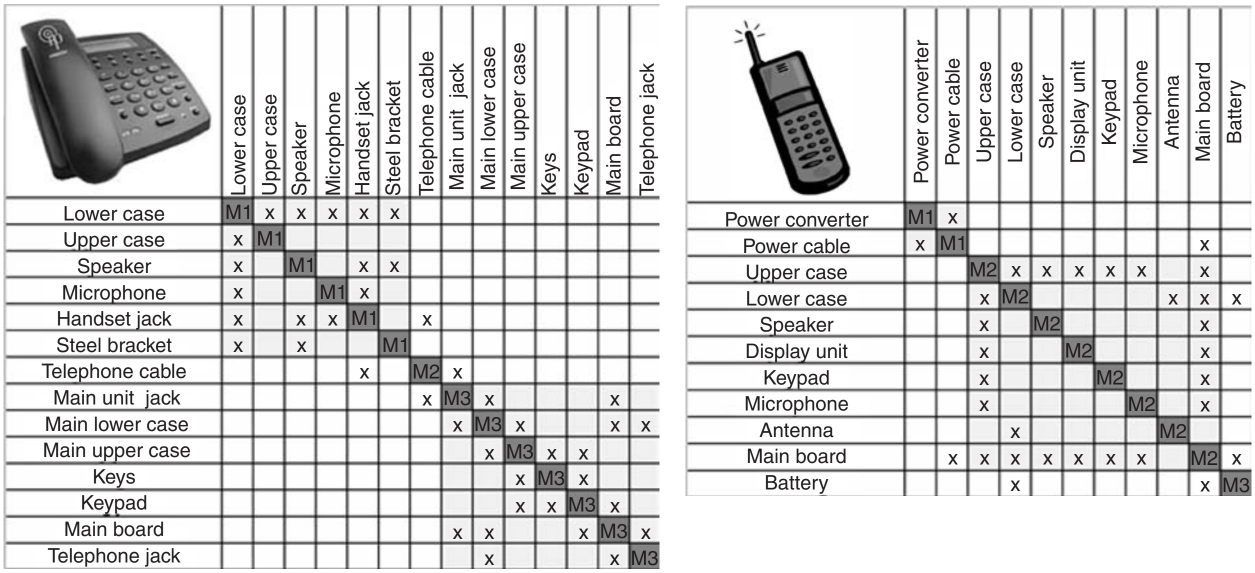
The DSM sparsity pattern for the four products that were decomposed is shown in Figure 6. This is also shown in Figure 8 as part of a larger set of 15 products as products 1, 6, 7, and 10, respectively.

Two modularity metrics were then calculated, the sparsity metric (NZF) and the packaging factors for each product. The modularity metrics include: Guo and Gershenson's modularity metric (Equation (1)), and the SMI (Equation (6)). Table 2 shows the results for all four products. The singular value decay patterns for SMI are shown in Figure 7, along with their exponential approximations.

Both the SMI and the Guo and Gershenson metric (M_2) indicate that the static version of the product has a higher degree of modularity than its mobile counterpart. The difference is more significant for the telephones than it is for the computers, when considering the SMI; it is the other way round when looking at M_2. The DSMs of the mobile products, i.e., those subject to more stringent technical constraints, are more densely populated (NZF is larger), and the amount of unused volume is also significantly smaller (higher packing density). The absolute difference between the modularity metric values in Table 2 appears to be relatively small, whereas the same difference between products in the same pair is seen more clearly in Figure 7.

The results shown in Figure 7 and Table 2 are in line with the hypothesis that technical constraints will have a tendency to drive designs towards more integral architectures. Similar results can be seen with the

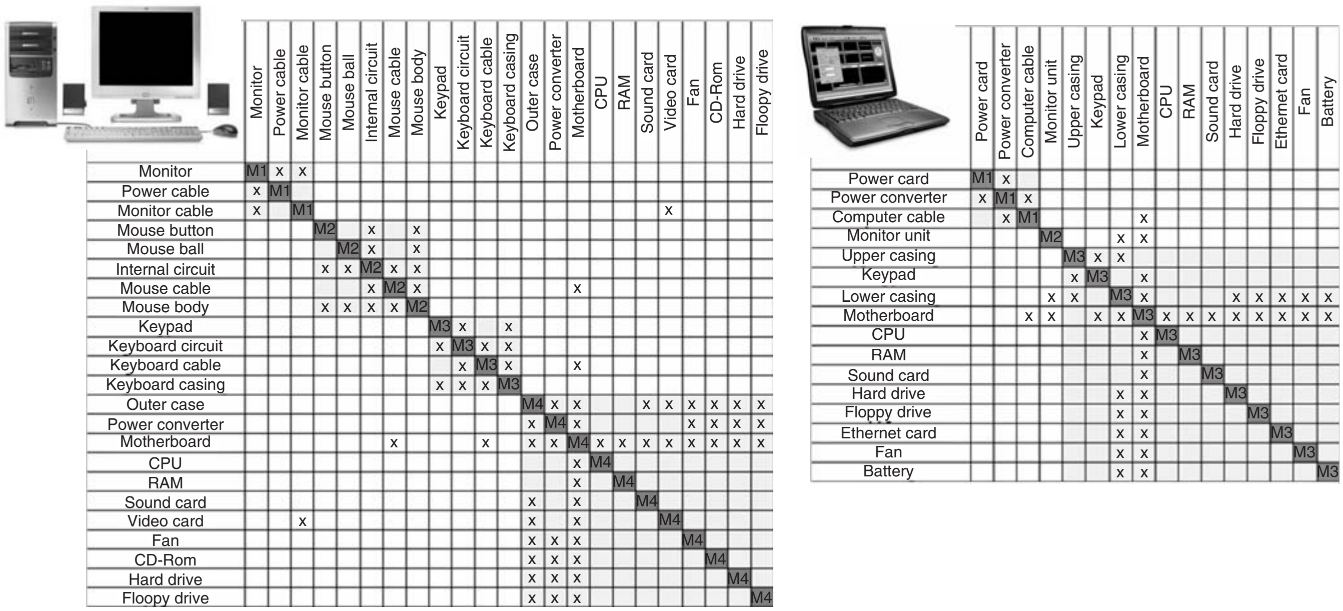
Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.



	Lower case	Upper case	Speaker	Microphone	Handset jack	Steel bracket	Telephone cable	Main unit jack	Main lower case	Main upper case	Keys	Keypad	Main board	Telephone jack
Lower case	M1	x	x	x	x	x								
Upper case	x	M1												
Speaker	x		M1		x	x								
Microphone	x			M1	x									
Handset jack	x		x	x	M1		x							
Steel bracket	x		x			M1								
Telephone cable					x		M2	x						
Main unit jack							x	M3	x				x	
Main lower case								x	M3	x			x	x
Main upper case									x	M3	x	x		
Keys										x	M3	x		
Keypad										x	x	M3	x	
Main board								x	x			x	M3	x
Telephone jack									x				x	M3

	Power converter	Power cable	Upper case	Lower case	Speaker	Display unit	Keypad	Microphone	Antenna	Main board	Battery
Power converter	M1	x									
Power cable	x	M1								x	
Upper case			M2	x	x	x	x	x		x	
Lower case			x	M2					x	x	x
Speaker			x		M2					x	
Display unit			x			M2				x	
Keypad			x				M2			x	
Microphone			x					M2		x	
Antenna				x					M2		
Main board		x	x	x	x	x	x	x		M2	x
Battery				x						x	M3

Figure 4. Component-to-component design structure matrix for two telephones: desk phone (left), cellular phone (right), the cellular phone is subject to volume (packaging) and lightweighting constraints.



	Monitor	Power cable	Monitor cable	Mouse button	Mouse ball	Internal circuit	Mouse cable	Mouse body	Keypad	Keyboard circuit	Keyboard cable	Keyboard casing	Outer case	Power converter	Motherboard	CPU	RAM	Sound card	Video card	Fan	CD-Rom	Hard drive	Floopy drive
Monitor	M1	x	x																				
Power cable	x	M1																					
Monitor cable	x		M1																x				
Mouse button				M2		x		x															
Mouse ball					M2	x		x															
Internal circuit				x	x	M2	x	x															
Mouse cable						x	M2	x							x								
Mouse body				x	x	x	x	M2															
Keypad									M3	x		x											
Keyboard circuit									x	M3	x	x											
Keyboard cable										x	M3	x			x								
Keyboard casing									x	x	x	M3											
Outer case													M4	x	x			x	x	x	x	x	x
Power converter													x	M4	x					x	x	x	x
Motherboard							x				x		x	x	M4	x	x	x	x	x	x	x	x
CPU															x	M4							
RAM															x		M4						
Sound card													x		x			M4					
Video card			x										x		x				M4				
Fan													x	x	x					M4			
CD-Rom													x	x	x						M4		
Hard drive													x	x	x							M4	
Floopy drive													x	x	x								M4

	Power card	Power converter	Computer cable	Monitor unit	Upper casing	Keypad	Lower casing	Motherboard	CPU	RAM	Sound card	Hard drive	Floppy drive	Ethernet card	Fan	Battery
Power card	M1	x														
Power converter	x	M1	x													
Computer cable		x	M1					x								
Monitor unit				M2			x	x								
Upper casing					M3	x	x									
Keypad					x	M3		x								
Lower casing				x	x		M3	x				x	x	x	x	x
Motherboard			x	x		x	x	M3	x	x	x	x	x	x	x	x
CPU								x	M3							
RAM								x		M3						
Sound card								x			M3					
Hard drive							x	x				M3				
Floppy drive							x	x					M3			
Ethernet card							x	x						M3		
Fan							x	x							M3	
Battery							x	x								M3

Figure 5. Component-to-component design structure matrix for two computers: desktop (left) and laptop (right).

Guo and Gershenson modularity metric. The two products with less stringent weight, power, or similar constraints and higher packaging factors (more unused volume) have a larger modularity index and are therefore more modular than their constrained counterparts. The indices only differ by modest amounts, but if one looks at the connectivity between modules (the latter half of the Guo and Gershenson metric (Equation (1)) for both product pairs, the connectivity in both laptop (0.15) and cellular phone (0.25) is clearly larger than in desktop PC (0.03) and desk phone (0.13), respectively, also indicating that the cellular phone is more integral than the desk phone, further supporting the hypothesis.

7. Modularity versus Integrality

The preliminary conclusion is that the hypothesis is supported and it is concluded that technical constraint-based design of systems and products typically leads to a high degree of coupling between elements of form (=integral architectures). This, however, needs to be further substantiated by considering a larger product set to see if more general conclusions can be drawn about the spectrum between integrality and modularity.

In addition to the efforts by the authors, DSM information has been collected about a number of

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

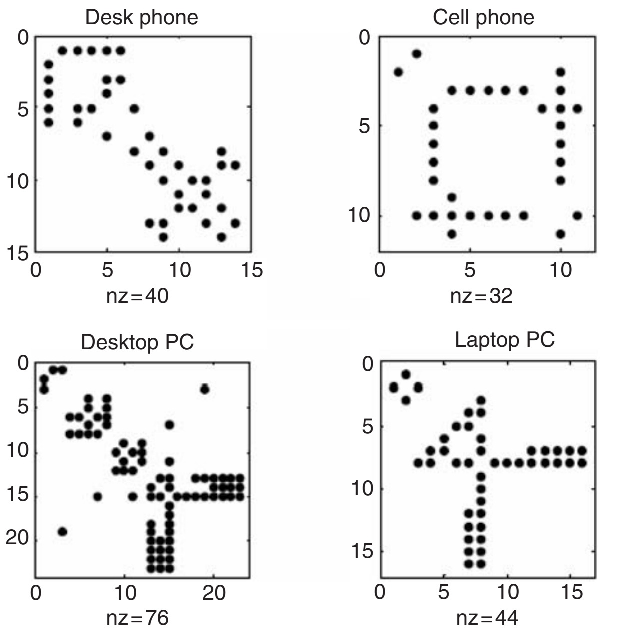


Figure 6. Sparsity pattern for static products (left) and mobile products (right).

products from the literature and a dataset has been assembled for 15 different systems and products. For each the binary DSM was generated based on either the existing DSM or a function structure that was available, in a standardized format and the SMI and NZF metrics were calculated. The other metrics could not be calculated for these products due to lack of available data.

The sparsity patterns of the 15 products are shown in Figure 8. Note that the systems are numbered from 1 to 15 in the alphabetical order. The source of the data is indicated in the caption of the figure by referring to the appropriate paper in the bibliography. The systems range in size from $N = 10$ (intra-oral camera [31]) to $N = 54$ (jet engine [17]) with an average size of $N(\text{mean}) = 21$ elements. The mean sparsity is $\mu(\text{NZF}) = 0.15$ with a standard deviation of $\sigma(\text{NZF}) = 0.075$. The SMI was computed for all systems and the system with the lowest $\text{SMI} = 0.04$ is the idealized picket fence. A picket fence's architecture is very similar to the structure of a bus-modular system (Figure 2(b)) with the exception that there are two fence

Table 2. Product name, DSM size, packaging factor, sparsity, and two measures of modularity for two functionally equivalent product pairs: desktop PC-laptop and desk phone-cellular phone.

Product	N	P (Equation (2))	NZF (Equation (7))	M_2 (Equation (1))	SMI (Equation (6))
Desk phone	14	0.39	0.21	0.59	0.45
Cell phone	11	0.10	0.29	0.57	0.23
Desktop PC	23	0.82	0.15	0.68	0.24
Laptop	16	0.10	0.18	0.55	0.20

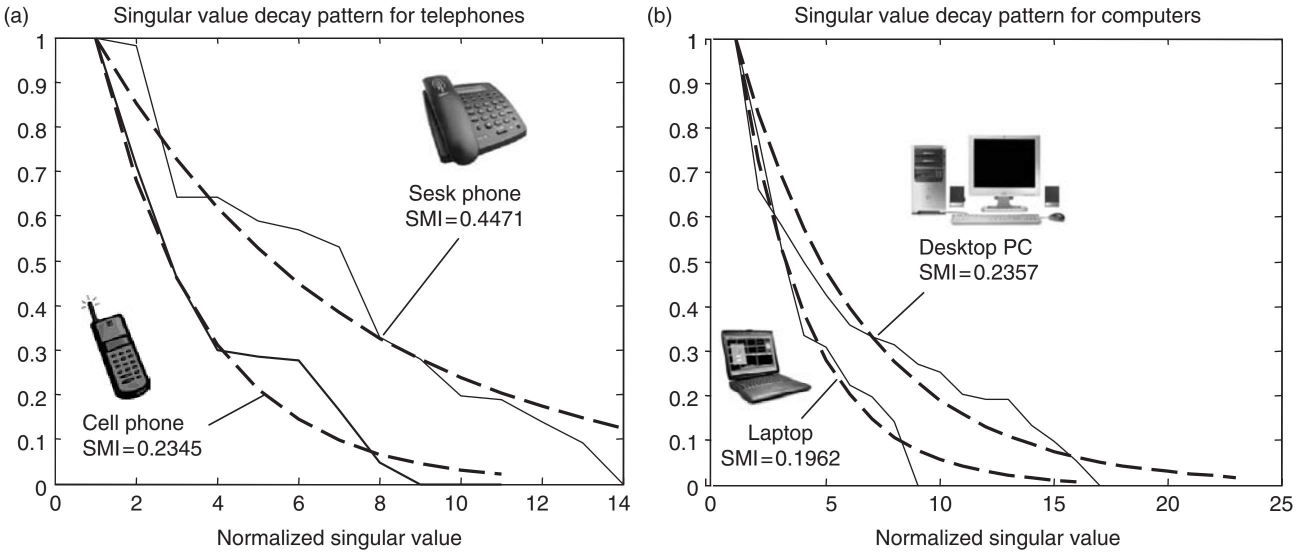


Figure 7. Singular value decay patterns for the pairs of telephones (left) and computers (right). A slower decay pattern (tendency towards the upper right) is indicative of a more modular architecture.

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

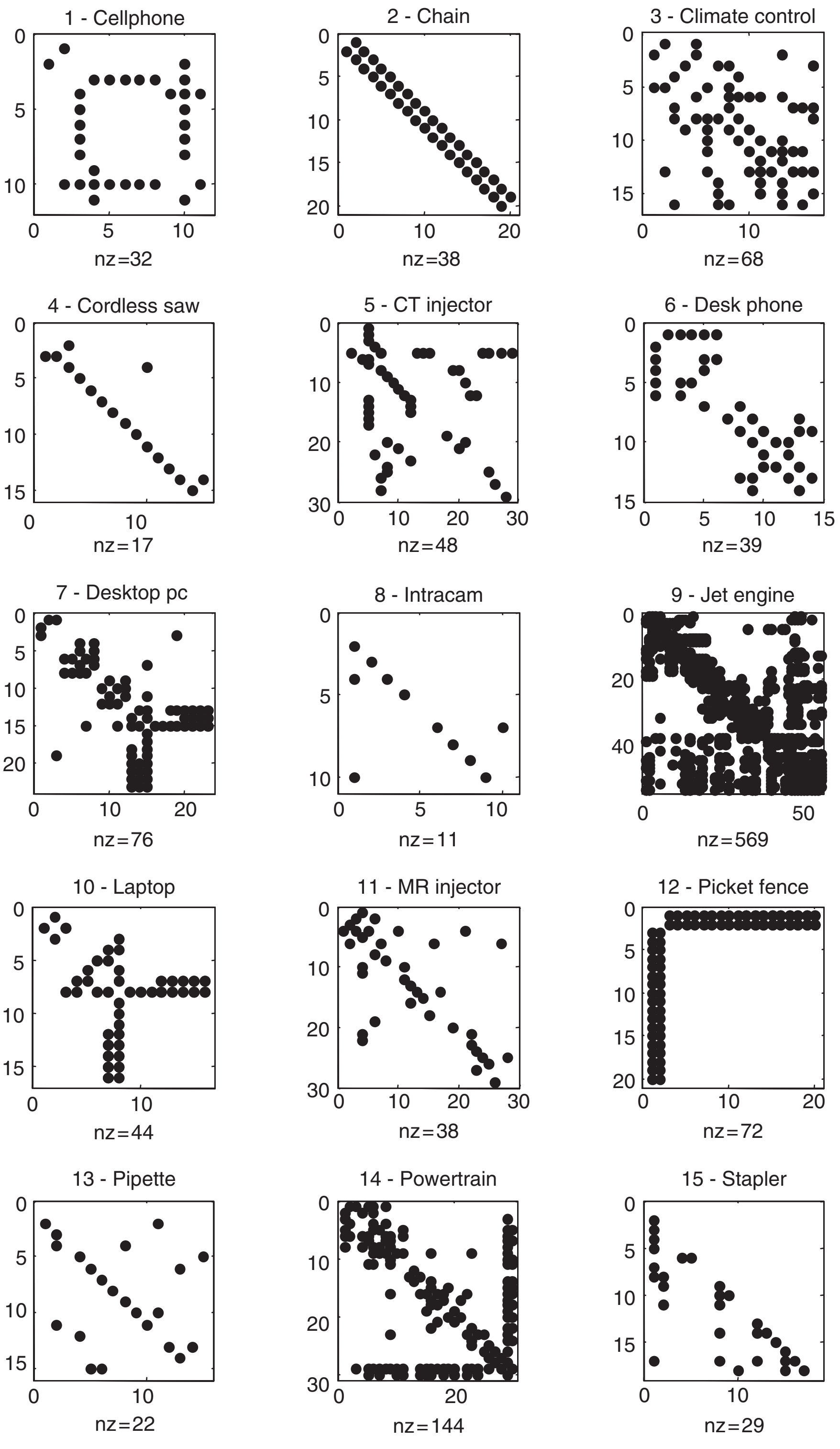


Figure 8. Sparsity patterns of binary DSM for 15 different systems and products: 1- cell phone, 2- chain, 3- climate control system [10], 4- cordless saw [33], 5- CT injector [32], 6- desk phone, 7-desktop PC, 8- intraoral camera [31], 9- jet engine [17], 10- laptop, 11- MR injector (unpublished sample), 12- picket fence, 13- electronic pipette [31], 14- powertrain [34], 15- stapler [35].

rails connected to each slat. The system with the highest modularity (SMI = 0.95) is a chain, where each link only connects to its direct neighbors (see Figure 2(c)). The mean value of SMI across the 15 systems is 0.39. A visual inspection of Figure 8 shows that the jet engine clearly stands out in terms of architectural complexity.

In Figure 9, these 15 systems are plotted in terms of SMI versus NZF, i.e., modularity versus sparsity.

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

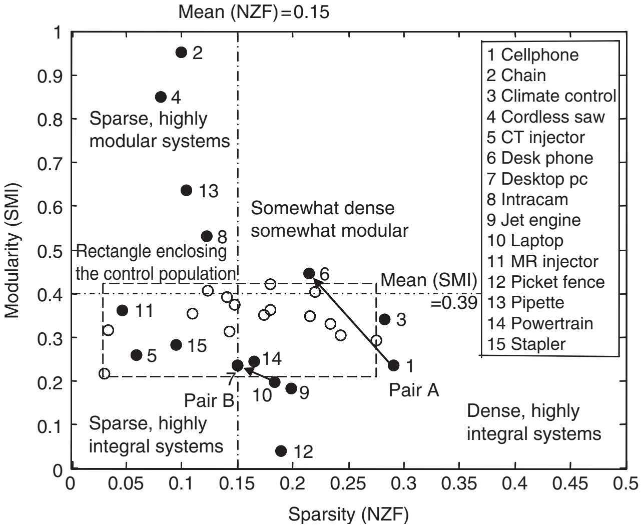


Figure 9. Modularity (SMI) versus sparsity (NZF) for 15 products and systems, plotted against a control group of 15 randomly generated systems (empty circles, enclosed by dashed box) with the same mean and standard deviation in terms of size (N) and sparsity (NZF).

The average density of this population (mean of NZF) is 0.15, while the average SMI is 0.39. The systems in the upper left exhibit higher than average modularity, but lower than average density; they are both highly modular, and sparse. The chain, cordless saw, and to some extent the pipette and intra-oral camera fall into this category. These are systems that can be made modular based on technical feasibility and business needs. In the lower right, are products that are very dense (many interconnections between parts), while also being very integral. The mobile products (cell phone, laptop), but also the power intensive systems (jet engine, power train, climate control) fall into this category. Most systems fall into the intermediate regime and are neither fully modular nor fully integral. Of the product pairs investigated in Section 5, both Pair A (telephones 1–6) and Pair B (computers 7–10) show that the static version of the product has a less dense architecture as well as a somewhat higher degree of modularity.

The significance of these results is corroborated by generating a population of 15 random DSMs whose mean and standard deviation in terms of size (*N*) and sparsity (NZF) is the same as for the products shown in Figure 8. The DSMs of the random control population are shown in Appendix A. In the SMI versus NZF space all members of the random population are enclosed in the dashed box shown in Figure 9. This suggests that deliberately architected products, especially those that are very integral or very modular do not emerge randomly but are either driven by technical or business considerations. It is also noteworthy that there are no systems that are both dense (large NZF) and highly modular (large SMI) at the same time, since modularity by its very nature implies sparsity.

Figure 9 further supports Whitney's argument that high power systems tend to be more integral than low power systems. The high power examples (e.g., powertrain, jet engine) are at the lower end, the integral end, of the figure with an SMI < 0.35. Similarly the low-power products tend to be at the top end, modular end, of the figure. Whitney's argument however is not enough to explain why a cordless tool (cordless saw) for example is able to be so modular. Some of the differences come from the fact that the architectural representations used in the analysis for these 15 products were created by separate individuals, using different assumptions, and different levels of decomposition in terms of individual components and interconnections (physical, material, power, information). An example of this is the DSM of the stapler (lower right, system #15) in Figure 8 which is only being reported as a single-sided DSM. This impacts the fidelity of the DSMs, but not the repeatability of the SMI computations. This will be discussed further in the next section. More in-depth analysis is needed to determine the full extent and validity of these results.

8. Conclusions and Discussion

The hypothesis is supported in that designs that are heavily driven by technical constraints (weight, volume, power) tend to exhibit more integral architectures than those that do not have these constraints. In order to show this at first, the literature was surveyed and was found that a number of references discuss the tension between modularity and integrality in technical design. Then a new modularity metric was developed, the singular value modularity index (SMI) which measures the rate of decay of the singular values (square roots of the eigenvalues of $\mathrm{DSM}^{\mathrm{T}}\mathrm{DSM}$) of a binary system DSM. The faster the eigenvalues decay, the smaller the SMI and the less modular a particular system architecture is.

The SMI was developed using a set of three canonical architectures (integral, bus-modular, and modular) and applied it to two sets of products (telephones, computers) which had been previously reverse engineered. The results showed that the mobile, lightweight, and compact versions of these otherwise functionally equivalent products are indeed more integral. Efficiency is gained by having fewer components and by assigning more than one function to each component. While the differences in SMI values were relatively modest, 0.45 versus 0.23 for the telephones and 0.24 versus 0.20 for the computers, the results were corroborated by the modularity metric developed by Guo and Gershenson [28] as well as by visual inspection of the singular value

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

decay patterns. The advantage of SMI over previously proposed modularity metrics is that it (i) does not depend on subjective definition of module boundaries and (ii) always returns the same value regardless of the ordering of rows and columns of the DSM.

It is also showed how the size, weight, and power constraints can be represented quantitatively using metrics, such as the packaging factor. A small packaging factor indicated that most of the available space is used in the product, which naturally leads to a smaller sized and oftentimes more integral product. This particular point will require significantly more research, but holds the promise of turning modularity metrics, such as SMI, into a design tool. Specifically, if non-dimensional relationships between sparseness, modularity, and technical constraints (weight, packaging, power density) could be found, then one could assess the achievable technical performance of systems based on their underlying (modular or integral) architecture, or, conversely, one could estimate the achievable degree of modularity based on technical performance targets and constraints. This, however will require a much larger dataset. Furthermore, such relationships would likely evolve dynamically due to technological innovation (e.g., miniaturization, executing functionality in software rather than in hardware and so forth).

The results presented here are in apparent violation of the independence axiom in axiomatic design as well as of the idea that a high degree of modularity is always achievable or is always a virtue. It is shown that, at least in certain cases, an axiomatic or modular design does not emerge based on technical constraints. The insight is that in situations where performance is critical, other design methodologies than axiomatic or modular design also need to be considered.

Future work will focus on collecting larger datasets and assessing the robustness of SMI against DSMs for the same system at different levels of detail and for different modeling assumptions.

References

1. Ulrich, K.T. (1995). The Role of Product Architecture in the Manufacturing Firm, *Research Policy*, **24**(3): 419–441.
2. Suh, N.P. (2000). *Axiomatic Design: Advances and Applications*, New York: Oxford University Press, p. 528.
3. Whitney, D.E. (2004). Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development, New York: Oxford University Press, p. 544.
4. Boeing, L.R. – BWB technical paper.
5. Ericsson, A. and Erixon, G. (1999). *Controlling Design Variants*, New York: ASME Press, p. 146.
6. Fujita, K. and Yoshida, H. (2004). Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes, *Concurrent Engineering: Research and Applications*, **12**(2): 105–118.
7. Jiao, R., Huang, G. and Tseng, M. (2004). Concurrent Enterprising for Mass Customization, *Concurrent Engineering: Research and Applications*, **12**(2): 83–88.
8. Brabazon, P. and MacCarthy, B. (2004). Virtual-Build-to-Order as Mass Customization Order Fulfillment Model, *Concurrent Engineering: Research and Applications*, **12**(2): 155–165.
9. Newcomb, P.J., Bras, B. and Rosen, D.W. (1998). Implications of Modularity on Product Design for the Life Cycle, *Journal of Mechanical Design*, **120**(3): 483–490.
10. Pimmler, T.U. and Eppinger, S.D. (1994). Integration Analysis of Product Decompositions, *ASME Design Engineering Technical Conferences*, pp. 343–351, Minneapolis, MN.
11. Stone, R.B., Wood, K.L. and Crawford, R.H (2000). A Heuristic Method for Identifying Modules for Product Architecture, *Design Studies*, **21**(1): 5–31.
12. Zamirowski, E.J. and Otto, K.N. (1999). Identifying Product Family Architecture Modularity Using Function and Variety Heuristics, *ASME Design Engineering Technical Conferences*, Las Vegas, NV.
13. Holtta, K. and Salonen, M. (2003). Comparing Three Modularity Methods, *ASME Design Engineering Technical Conferences*, Chicago, IL.
14. Kusiak, A. and Huang, C. (1996). Development of Modular Products, *IEEE Transactions on Components, Packaging and Manufacturing Technology*, Part A, **19**(4): 523–538.
15. Aarnio, J. (2003). *Modularization by Integration: Creating Modular Concepts for Mechatronic Products*, PhD Thesis, Tampere University of Technology.
16. Braha, D. (2002). Partitioning Tasks to Product Development Teams, *ASME Design Engineering Technical Conferences*, Montreal, Canada.
17. Sosa, M.E., Eppinger, S.D. and Rowles, C.M. (2003). Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions, *Journal of Mechanical Design*, **125**, Part 2: 240–252.
18. Browning, T.R. (2001). Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions, *IEEE Transactions on Engineering Management*, **48**(3): 292–306.
19. Eppinger, S.D. (2001). Innovation at the Speed of Information, *Harvard Business Review*, **79**(1): 149–158.
20. Fixson, S.K. and Clark, J.P. (2002). On the Link Between Modularity and Cost – Methodology to Assess Cost Implications of Product Architecture Differences, *IEEE International Engineering Management Conference (IEMC 2002)*. Cambridge, UK, pp. 131–136.
21. Gonzalez-Zugasti, J.P. and Otto, K.N. (2000). Platform-Based Spacecraft Design: A Formulation and Implementation Procedure, *IEEE Aerospace Conference*, Vol. 1, pp. 455–463.

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

22. Whitney, D.E. (2004). *Physical Limits to Modularity*, Working Paper, ESD-WP-2003-01.03-ESD, Massachusetts Institute of Technology, Engineering Systems Division.
23. Cutherell, D. (1996). Product Architecture, In: Rosenau, M., Griffin, A., Castellion, G. and Anschuetz, N. (eds), *The PDMA Handbook of New Product Development*, New York: John Wiley & Sons, pp. 217–235.
24. Benini, L. and de Micheli, G. (1999). System-Level Power Optimization: Techniques and Tools, *Int. Symp. Low-Power Electronics Design*, pp. 288–293, San Diego, CA.
25. Kazman, R., Barbacci, M., Klein, M., Carrière, S.J. and Woods, S.G. (1999). Experience with Performing Architecture Tradeoff Analysis, *International Conference on Software Engineering*, Los Angeles, CA.
26. Bass, L., Clements, P. and Kazman, R. (2003). *Software Architecture in Practice*, 2nd edn, Addison-Wesley, Boston, MA.
27. Dong, Q. and Whitney, D. (2001). Designing a Requirement Driven Product Development Process, *ASME Design Engineering Technical Conferences*, Pittsburgh, PA.
28. Guo, F. and Gershenson, J.K. (2004). A Comparison of Modular Product Design Methods on Improvement and Iteration, *ASME Design Engineering Technical Conferences*, Salt Lake City, UT.
29. Holtta, K., Suh, E.S. and de Weck, O. (2005). Tradeoff Between Modularity and Performance for Engineered Systems and Products, *International Conference on Engineering Design (ICED05)*, Melbourne, Australia.
30. Antoulas, A.C., Sorensen, D.C. and Zhou, Y. (2002). On the Decay Rate of Hankel Singular Values and Related Issues, *Systems and Control Letters*, **5**(46): 323–342.
31. Holtta, K. (2002). Identifying Common Modules for Collaborative R&D, *Meeting on Production and Operations Management*, San Francisco, CA.
32. Holtta, K. and Otto, K. (2003). Incorporating Design Complexity Measures in Architectural Assessment, *ASME Design Engineering Technical Conferences*, Chicago, IL.
33. Otto, K. and Wood, K. (2001). *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River, NJ.
34. Smaling, R. and de Weck, O.L. (2007). Assessing Risks and Opportunities of Technology Infusion in System Design, *Systems Engineering*, **10**:1.
35. Van Wie, M.J., Greer, J.L., Campbell, M.I., Stone, R.B. and Wood, K.L. (2001). Interfaces and Product Architecture, *ASME Design Engineering Technical Conferences*, Pittsburgh, PA.

Katja Hölttä-Otto

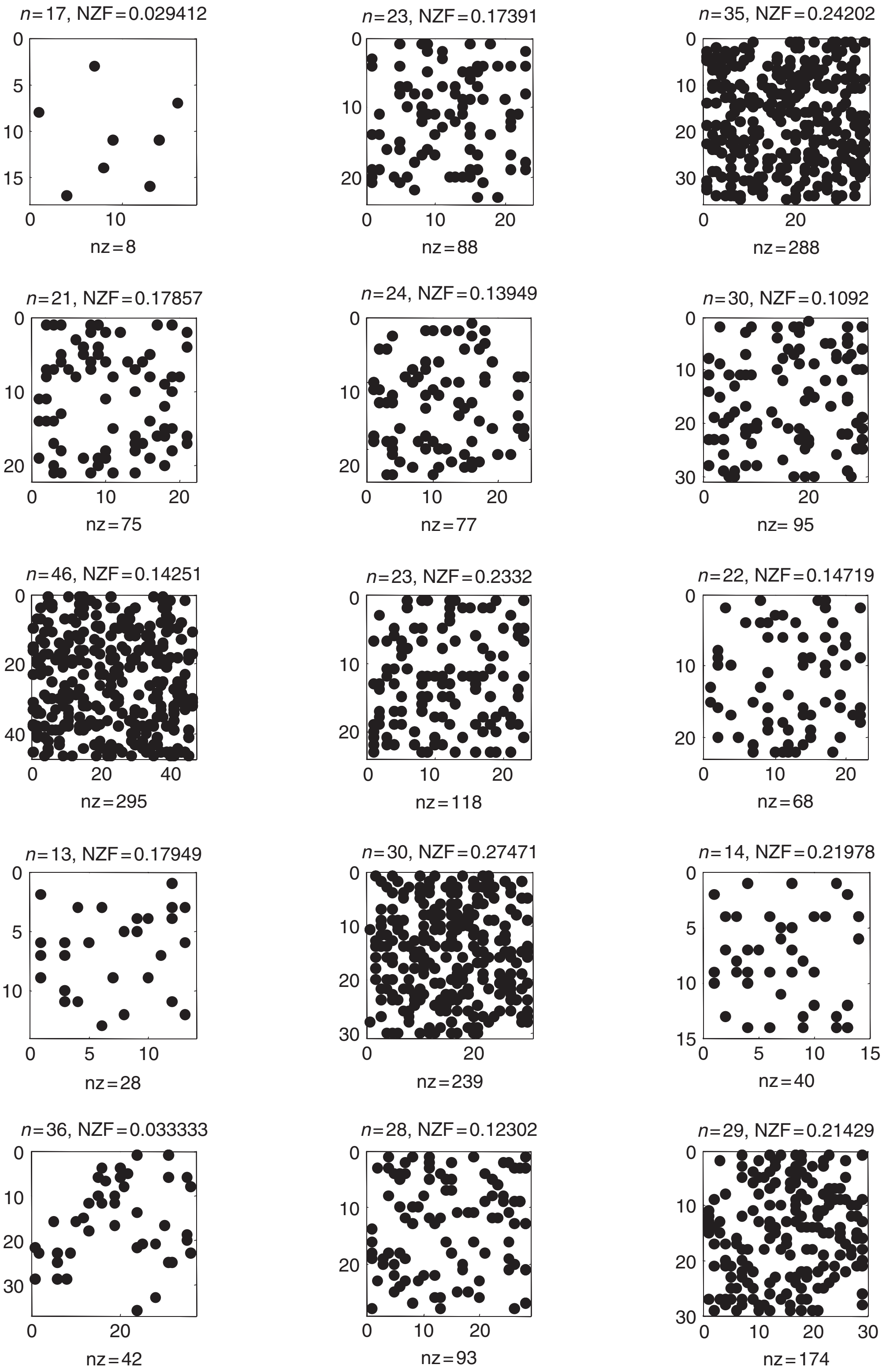
Katja Hölttä-Otto is an Assistant Professor of Mechanical Engineering at University of Massachusetts Dartmouth. In 2005, she received the degree of DSc in Mechanical Engineering from the Helsinki University of Technology, Finland. During her studies, she was a visiting scholar at MIT Center for Innovation in Product Development from 2001 to 2005. She obtained the best paper award at the ASME Design Engineering Technical conferences, conference on Design Theory and Methodology in 2006. Her research interests include product architecture and methods for product platform development.

Olivier L. de Weck

Olivier L. de Weck is an Associate Professor with a dual appointment between the Department of Aeronautics and Astronautics and the Engineering Systems Division at MIT. In 2001 he obtained a PhD in Aerospace Systems from MIT. From 1987 to 1993 he attended the Swiss Federal Institute of Technology (ETH Zurich), where he earned a Masters degree in industrial engineering. From 1993 to 1997 he served as liaison engineer and as engineering program manager at McDonnell Douglas (now Boeing). He obtained two best paper awards at the 2004 INCOSE Systems Engineering Conference. His research interests are in Multidisciplinary Design Optimization and Systems Engineering.

Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

Appendix A – DSMs of Randomly Generated Control Population



Downloaded from http://cer.sagepub.com by guest on August 3, 2007
© 2007 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.