INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15 – 18, 2005

TRADEOFF BETWEEN MODULARITY AND PERFORMANCE FOR ENGINEERED SYSTEMS AND PRODUCTS

Katja Hölttä, Eun Suk Suh, and Olivier de Weck

Keywords: Modularity, performance, trade-off, architecture

1 Introduction

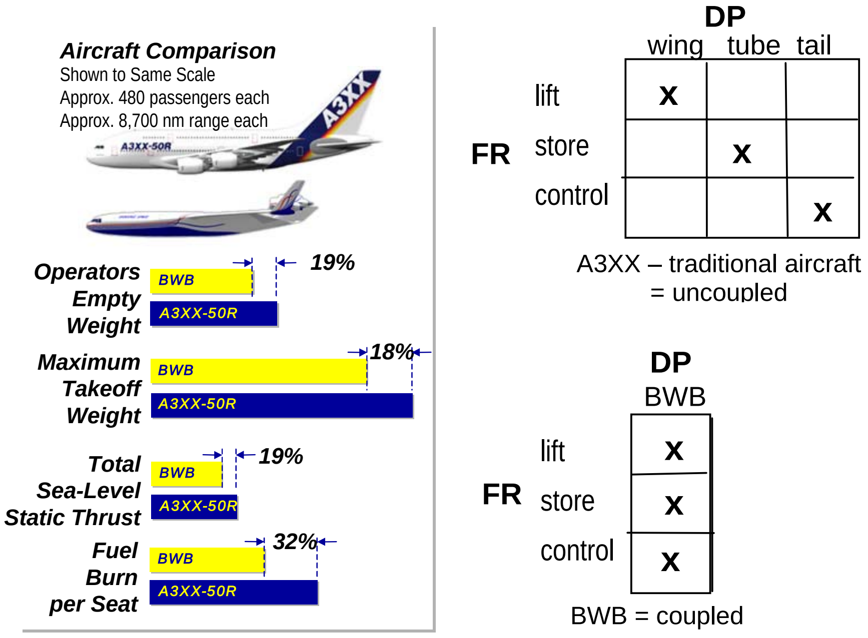
Modularity has become a pervasive theme in product development. Modular architecture, defined as having a one-to-one mapping from functional elements to the physical components of the product, or as “uncoupled design”, has many benefits from cost savings due to commonality to independent design of modules. However, a fully modular design may not always be achievable in designing engineering systems. In this paper, we show that technical constraints, such as light weighting, tight packaging and low power consumption, can drive designers towards more integral architectures. We quantify the degree of modularity by calculating three different modularity measures for two product pairs that represent business and technical constraint driven versions of the same product type. One of these metrics uses a singular value decomposition of the binary design structure matrix (DSM) as a measure of modularity.

2 Motivation

A module is often defined as a chunk that is loosely coupled to the rest of the system and similarly modular architecture is commonly defined as having a one-to-one mapping from functional elements in the function structure to the physical components of the product [23]. This is also supported by the independence axiom in axiomatic design [22]. The axiom states that functional requirements should be kept uncoupled with the design parameters. This independence has 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 (flexibility), increased robustness, and so forth.

For a number of systems this has been shown to be an effective approach to system design [22]. 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 performance efficiency for complex products and systems?

We suspect 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 (volume), mass and energy during operations. There is, generally a price to be paid for modularity in terms of other product characteristics. Whitney claims that modular architectures tend to have more parts, tend to be uncoupled or decoupled and favor “business performance". Integral architectures on the other hand have fewer parts, tend to be coupled and favor "technical performance" [26]. Some high performance systems such as automotive and aerospace vehicles appear to favor highly coupled architectures, where one part fulfills potentially many functions. An illustration is the new Blended Wing Body (BWB) concept developed by Boeing [17], see Figure 1. In this radical design the assignment of functionality between wing, fuselage and empennage are blurred, i.e. blended. A traditional "tube-and-wing" aircraft (e.g. A3XX, now A380) uses wings for lift generation, a cylindrical fuselage for storage of passengers and cargo as well as the tail empennage for generating control moments, primarily in pitch and yaw. In the BWB on the other hand the integral "blended" body provides all three functions to some extent (Fig.1b). 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 (Fig. 1a).



© Boeing

(a) (b)

Figure 1. (a) Comparison of traditional "wing-and-tube" aircraft and blended wing body (BWB) in terms of aircraft performance characteristics [17]. Image courtesy Boeing [17], (b) Notional comparison of system architectures in terms of their FR-DP Design Matrix (DM) [4].

In summary, there appears to be a potential trade-off between the desire for modularity from a "business" or lifecycle engineering standpoint and the desire for high performance and efficiency in the technical domain. As a consequence we can formulate the following hypothesis that, if proven correct, might be considered an important principle of System and Product Architecture:

Hypothesis (1)

Technical constraints in terms of mass, volumetric and power efficiency imposed on the design of engineering systems leads to a higher degree of coupling between elements of form and functional elements (= integral architectures) relative to systems of equivalent functionality where such constraints are substantially relaxed.

3 Related work

3.1 Architecture

Ulrich [23] defines architecture as follows: Product architecture is the scheme by which the function of a product is allocated to physical components. He further defines modular and integral architectures: ”A modular architecture includes a one-to-one mapping from functional elements in the function structure to the physical components of the product, and specifies de-coupled interfaces between components. An integral architecture includes a complex (non one-to-one) mapping from functional elements to physical components and/or coupled interfaces between components.“

1.1 Modularity

Modularity has many advantages. The purpose of modularity may be to obtain advantages in economies of scale and scope, economies in parts sourcing, and support for mass customization [2], [9], [25]. Modularity also eases the end-of-life processes such as recycling [18]. Further, modularity provides flexibility that allows for both multiple product variations and technology development by changing of modules without requiring changes in the overall product [9], [24]. These advantages are related to the “business performance” of a company as opposed to the “technical performance”.

Due to the many advantages of modularity, many methods to modularize have been developed [9], [19], [21], [27]. Interestingly, Holtta and Salonen showed [13] that these methods give different suggestions for a modular architecture. As an example, these methods do not agree on the number of modules i.e. the degree of modularity (or integrality) of the architecture. The modular function deployment [9], for example, tends to lead toward more integral architecture due to its limit on the number of modules.

Kusiak [15] [16] presents yet another algorithm to develop modular multi-chip modules. He integrates the product and process design. His algorithm [15] has a step to define the upper bound for module size (level of integrality) but no clear rule how to determine the size. Aarnio [1] recently developed a method modularization by integration (MBI) that is a step toward managing the trade-off between modularity and integrality. His method takes functional and economical issues as well as product variety into account but even though functionality is thought of, performance is not a specific criterion. This is true in all methods because performance, weight, etc. are constraints, not design parameters and therefore hard to take into account.

All the methods have very different criteria on what basis to modularize. All are important, of course, but one cannot optimize perfectly according to all criteria. Technical performance is one important criterion in many industries and the above methods do not handle this aspect particularly well.

How to decide on the degree of modularity or on the number of modules? Braha [5] has proposed a method to decide on a team size based on a maximum number of attributes that a team can handle, and that can be used to decide the modules as well. Ericsson and Erixon [9] say the ideal number of modules is approximately the square root of the number of parts. This is based on minimizing the assembly time. These are two suggestions, but there is no clear way of defining the degree of modularity in a specific case.

1.2 Modularity vs. integrality

Sosa et al [20] introduce a method to identify whether a system is modular or integral based on the component interactions in the system DSM. 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 [23] points out that the PD process is different for modular and integral products. Sosa et al do not, however, say when a modular system is more appropriate than an integral system but rather how to design the process after the system type has been identified.

Fixson and Clark [10] present a 2 by 2 matrix with the number of components participating in a function on one axis and the total number of functions that components under consideration provide on the other. Modular and integral architectures are at the far ends of the matrix. Fixson and Clark talk about the effects of the architecture type to the costs along the supply chain. But they too do not say when to choose a modular architecture and when an integral one.

Ulrich [23] discusses many advantages of modularity. He also talks about technical performance. He says modularity helps optimize local performance characteristics. For global performance he says integral architecture is better especially if weight and size constraints. In addition he mentions natural frequency of vibration and electromagnetic radiation. Ulrich discusses the topic thoroughly, but only qualitatively. Also Gonzalez-Zugasti and Otto show that some performance is sacrificed to obtain goals of the individual products that are created for a platform. They claim that the ideal would be to give up some of performance locally in modules for a better platform. [11]

Whitney [25] has also written about how total modularity is not always desirable especially in case of high power mechanical products, as opposed to low power signal processor type products. A more modular product, according to Whitney [25], is likely to be larger, heavier, and less energy efficient. Also side effects are harder to control. He compares complex electro-mechanical-optical products to VSLI, which can be considered fully modular, and in line with Suh’s design axioms. 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 [25] also points out the interfaces (in high power systems) take out space and weight and they must be designed for each application. Also Benini and de Micheli [4] discuss the same issue. According to them power optimization is especially important in low power high performance systems (such as cellular phones). Benini and de Micheli discuss several methods for power optimization i.e. to handle the tradeoff between energy efficiency and flexibility.

According to Cutherell [6], integral architecture often driven by product performance or cost and modular architecture by variety, product change, engineering standards, and service requirements. He adds that performance is usually measured in efficiency, size, weight, and speed. As examples he mentions a heavier car having less mileage. He also uses a hand-held electronic calculator as an example of almost fully integral (excl. battery) architecture in order to make the product light weight.

Kazman et al [14] discuss the architectural tradeoffs in software design. They introduce a method to identify tradeoff points between system modifiability, availability, and performance in order to achieve wanted system quality. Also Bass et al [3] talk about the same method called Architecture Tradeoff Analysis Method (ATAM).

There is a lot of research 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 paper we attempt to show through two examples more quantitatively what effect technical performance (efficiency) constraints have on the degree of modularity of functionally equivalent products.

4 Approach

In this section, we describe the method of analysis used during research. The main method is to use a design structure matrix (DSM) to analyze the coupling and modularity of a system. Dong and Whitney [7] showed that Suh's DM and DSM are closely linked and that most couplings in the other are due to couplings in the other matrix. This results in a need to analyze one of the matrices and we choose the DSM.

The validity of hypothesis (1) can be probed by analyzing the off-diagonal terms in the DSM's of a large range of systems. We use modularity metrics from the past literature, but in addition we developed a new metric that we believe is less sensitive to human choices on how to draw module boundaries than one of the previous metrics [12].

An approach to proceed in order to ultimately support or refute the hypothesis is as follows:

1. Select a variety of systems/products in terms of business and technical performance. This includes pairs of products where one is a static, business-driven version, while the other is a mobile, performance-driven version of a functionally equivalent product.
2. Do a physical decomposition of the elements of form and of the functions of these selected systems.
3. Identify the modules and their interconnections by applying DSM.
4. Calculate modularity indices.
5. Discuss the commonalities and differences between the system architectures, particularly with respect to the degree of coupling.
6. Conclude by either supporting or refuting the hypothesis (1)

In the following section we introduce the new metric and then move on to analyze the different systems in the remainder of the paper.

5 Singular value Modularity Index (SMI)

We are seeking a way to quantify the degree of modularity of a product based on its internal connectivity structure. This structure can be represented by a binary design structure matrix (DSM), where the diagonal entries are zeros and off-diagonal elements are set to unity if two components are connected. This connection can be a physical connection or a transmission of power or information from one component to its connecting component.

Figure 2 shows idealized – and extreme – 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 and in the third case components only connect to their direct neighbor.

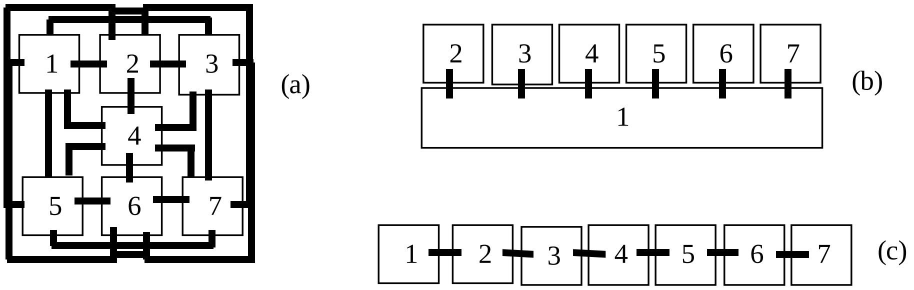


Figure 2. (a) Fully “integral” system, (b) “bus-modular” system , (c) fully “modular” system

The binary matrices for these systems are given as:

$$DSM_a = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} DSM_b = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} DSM_c = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (1)$$

These matrices can also be visualized in terms of their sparsity pattern (Fig.3).

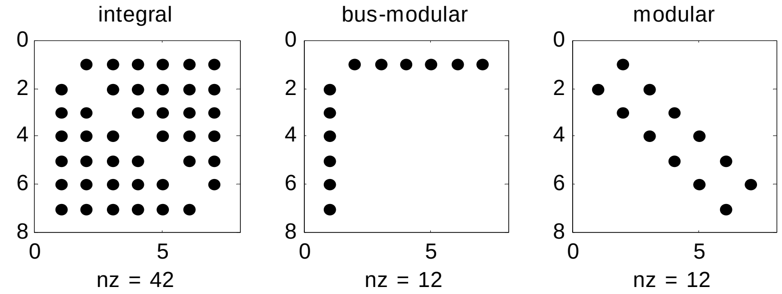


Figure 3: Sparsity pattern for idealized DSM’s: (a) fully integral, (b) bus-modular and (c) modular

Performing a Singular Value Decomposition (SVD) on the binary DSM matrix, reveals its singular values and corresponding orthogonal eigenvectors:

$$DSM = U \cdot \Sigma_{DSM} \cdot V^T \quad (2)$$

where

$$\Sigma_{DSM} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_N \end{bmatrix} \quad (3)$$

and the singular values, $\sigma_1, \sigma_2, \ldots, \sigma_N$ are ordered in decreasing magnitude and N is the number of components (= number of rows and columns in the DSM) of the system.

Figure 4 contains a plot of the singular values for the three systems (Fig.2a-c) shown in decreasing order of magnitude, respectively.

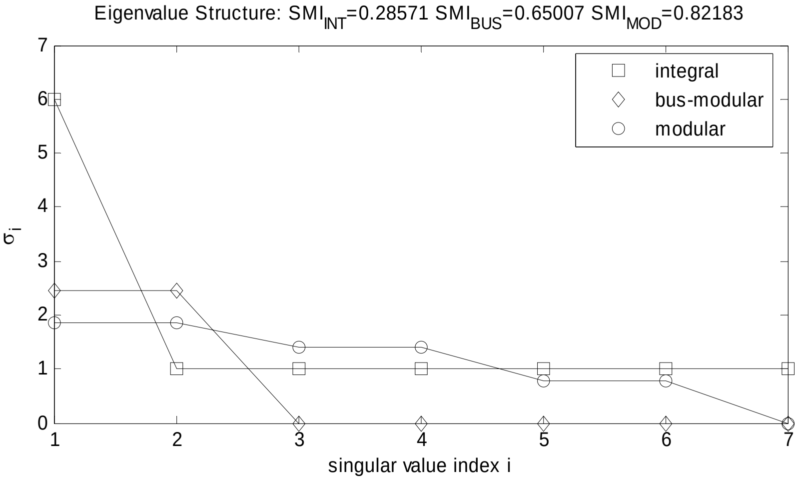


Figure 4: Singular values of “integral”, “bus-modular” and “modular” systems

There is a significant difference in the singular value decay structure. The integral system has one large singular value (6.0) followed by N-1 smaller ones (1.0), the bus-modular system has two large non-zero singular values (2.45) 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 Eq. (2). In other words, the information content to describe the connectivity of the system is different for modular versus integral systems. The modular design, for example requires more information to be described completely, relative to the bus-modular system. In other words, all singular values except one are non-zero and *N-1* eigenvectors must be retained for a complete description of the modular system.

We therefore postulate a modularity index that reflects the degree of to which the important information to describe the system is concentrated in a few highly connected components. Such systems show a much more rapid drop-off in the magnitude of singular values, relative to modular systems where such information is more widely distributed throughout the system. The singular value modularity index (SMI) therefore measures the average, weighted decay rate of sorted singular values in the system:

$$SMI\left(\Sigma_{DSM}\right) = 1 - \frac{1}{N \cdot \sigma_1} \sum_{i=1}^{N-1} \sigma_i \left(\sigma_i - \sigma_{i+1}\right) \tag{4}$$

This index is theoretically bounded between 0 and 1. An SMI closer to 1.0 indicates a higher degree of modularity, where the connectivity information is more broadly distributed. An SMI closer to 0 indicates a more integral system. The SMI for the idealized cases (Fig.3) is as follows: the “integral” design has an SMI_{INT}=0.29, the SMI for the “bus-modular” system is 0.65, the index for the “fully modular” design is SMI_{MOD}=0.82.

6 Other modularity metrics

Based on Ulrich’s definition of modular architecture having one to one mapping between functions and components, modular structures should have a smaller function to component ratio than integral products. The modularity metric according to Ulrich is simply:

$$N_{comp} / N_{funct} \tag{5}$$

The function-to-component ratio for each product we analyzed is shown in Table 1 below. Guo and Gershenson [12] developed a metric to measure product modularity using a component-to-component connectivity matrix. This 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:

$$\frac{\sum_{k=1}^{M}\frac{\sum_{i=n_k}^{m_k}\sum_{j=n_k}^{m_k}R_{ij}}{(m_k-n_k+1)^2}-\sum_{k=1}^{M}\frac{\sum_{i=n_k}^{m_k}(\sum_{j=1}^{n_k-1}R_{ij}+\sum_{j=m_k+1}^{N}R_{ij})}{(m_k-n_k+1)(N-m_k+n_k-1)}}{M} \tag{6}$$

where

n_k = index of the first component in k^{th} module

m_k = index of the last component in the k^{th} module

M = total number of modules in the product

N = total number of components in the product

R_{ij} = the value of the i^{th} row and j^{th} column element in the modularity matrix.

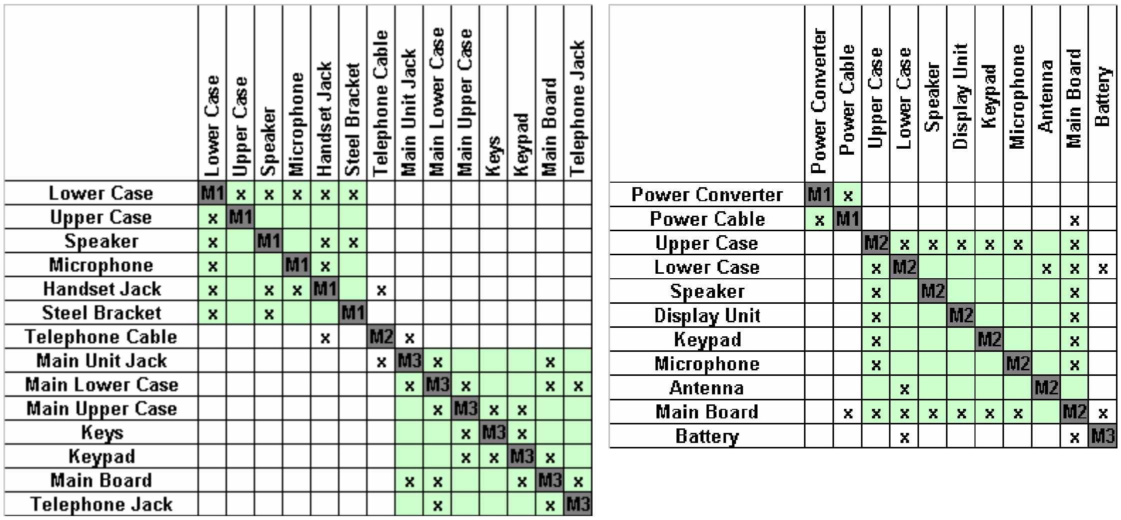
This metric can be applied to the component-to-component DSMs and the results are shown for the products we analyzed in Table 1. We apply the metric here, but we also observed during our research that the metric in Eq. (6) is sensitive to the choice of module boundaries. In analyzing existing products, we find the definition of a module to be subjective and found the SMI to be less sensitive to human choice. The Guo and Gershenson metric, however, measures the connectivity within modules and between modules and is useful when module boundaries are unambiguous.

7 Product dissection and DSM analysis

We chose products in pairs so that they shared similar functionality but were different in terms of business and technical performance. We include two pairs of products in the analysis: a cellular and desk telephone as well as a laptop and desktop computer (step 1). We continued by decomposing the elements of form and of the function of these selected systems down to approximately two levels of decomposition (step 2). Next, we assemble the function-form (DM) and component-to-component DSMs for each system (step 3). In order for the functionality to be as similar as possible between the two phones, we chose an older cellular phone (OKI, 1998) that did not include extra functionality compared to the desk phone (Sony, IT-B3, 1996). The cellular phone represents a product, where performance 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. For the computers we choose a Dell Desktop PC (Dimension Series Pentium II, 1999) and a Dell Inspiron 3800 (2000), respectively. Figure 5 shows pictorially product structure as determined during product dissection for the cellular telephone.

Figure 5: Product dissection example (cellular phone, OKI, 1998)

We decomposed the two phones and derived the following DSMs for them (Figure 6).



	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

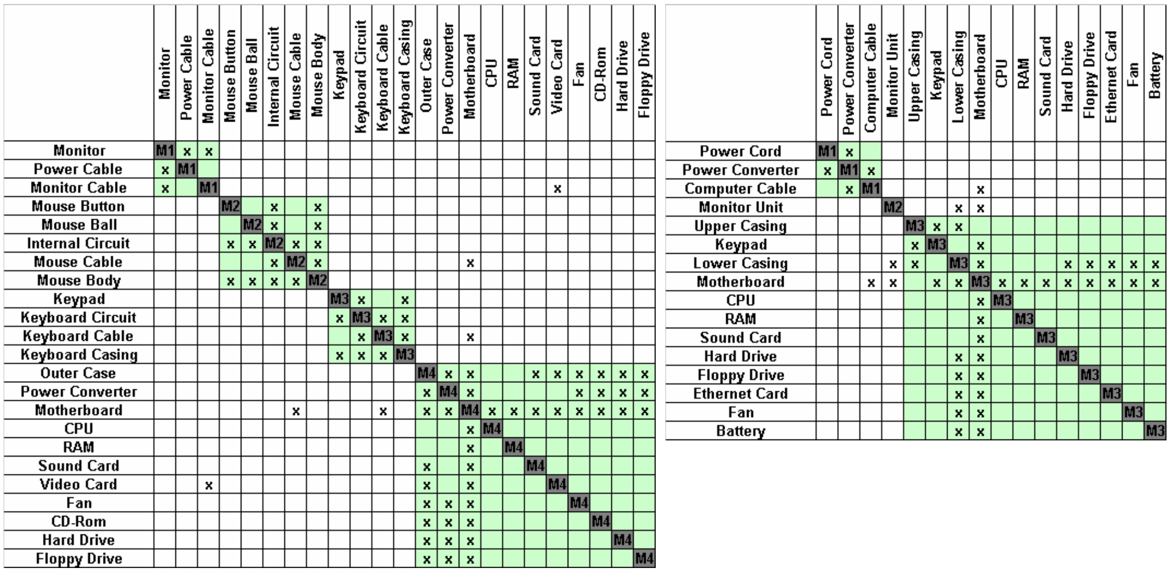
Desk phone

	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

Cellular phone

Figure 6. Component-to-component design structure matrix for the two phones

The second pair of products is a pair of computers. Here a laptop represents a product where design technical constraints determined the architecture during design (light weight, small volume, power efficiency). The desktop PC, on the other hand, is designed from business performance point of view (e.g. flexible to change, separate development). We decomposed the two computers and derived the following DSMs for them (Figure 7).



	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	Floppy 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	
Floppy Drive													x	x	x								M4

Desktop PC

	Power Cord	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 Cord	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

Laptop

Figure 7. Component-to-component design structure matrix for the two computers

The DSM sparsity pattern for the four products that were decomposed is shown in Figure 8.

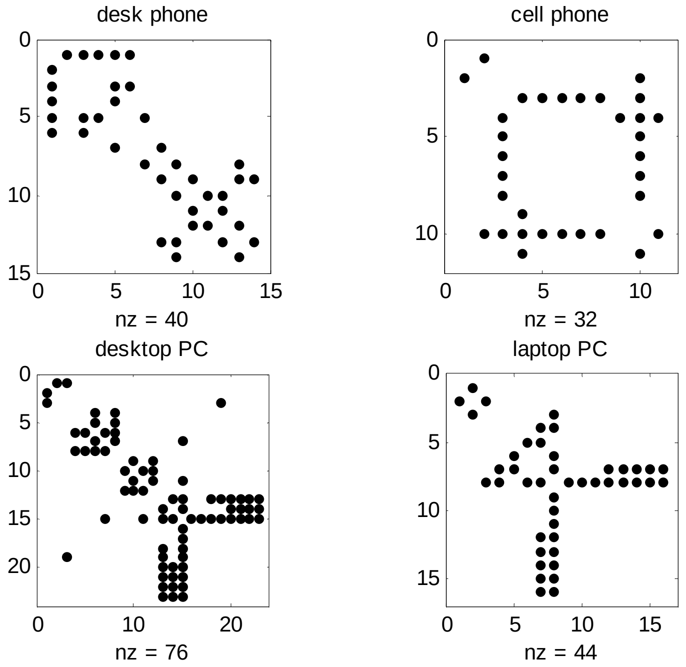


Figure 8. Sparsity patter for static products (left) and mobile products (right)

A visual inspection reveals that both the cell phone and the laptop computers feature one component that connects virtually across the entire product and resembles the bus-modular architecture of component #1 shown in Fig. 2b. For the cell phone this is component #3 (“upper case”), for the laptop it is component # 8 (“motherboard”). For each of these products an SVD of the binary DSM is performed (Eq. 2) and the results are shown in Fig. 9.

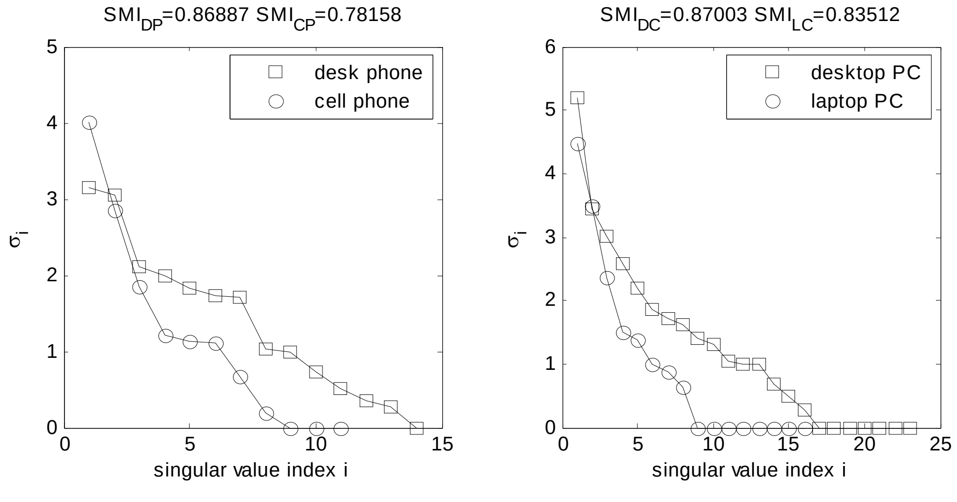


Figure 9. Singular value ranking for telephones (left) and computers (right).

Even though the static-vs-mobile versions of each product fulfill essentially the same external function, their internal structure is quite different. Inspecting Fig. 8 and Fig. 9 we recognize that the desk phone appears to be more modular than the cell phone, similarly the desktop appears significantly more modular than the laptop computer.

In order to quantify the degree of modularity we apply the modularity metrics discussed in Sections 5 and 6 to the DSMs shown in Fig. 6 and Fig.7, respectively. The results are summarized in Table 1. Since there is no clear agreement in the literature how to calculate the level of modularity of a product, we calculate modularity using the three different metrics in order to ascertain whether or not there is consistency in the results (step 4).

Table 1. Module index summary for desktop PC-laptop and desk phone-cellular phone pairs

	Performance properties/ constraints	SMI Eq. (4)	functions per components Eq. (5)	Guo and Gerhenson (0-1, 1 most modular), Eq. (6)
Desk phone		0.87	4.07	0.59
Cell phone	Compact Light weight Mobile	0.78	4.64	0.57
Desktop PC		0.87	4.43	0.68
Laptop	Compact Light weight Mobile	0.83	6.00	0.55

We proceed with an analysis of the results (step 5). As we can see in Table 1, the products with technical performance constraints (e.g. light weight, tight packaging, power efficiency) tend to have a larger functions-to-component ratio, i.e. they are more integral. Both static products, i.e., desk phone, and desktop PC have a lower functions-to-components ratio. The components in the technical constraint driven products are more coupled than in the business factor driven products as hypothesized

Similar results can be seen with the Guo & Gershenson modularity metric. The two products with no weight, power, or similar constraints have a larger modularity index and are therefore more modular than their constrained counterparts. The index differs only a little in the phones, but if one looks at the connectivity between modules (the latter half of the Guo & Gershenson metric) 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, indicating that also the cellular phone is more integral than the desk phone, further supporting our hypothesis.

The SMI metric suggests that the static products, desk phone and desktop PC, are similar – but not identical- to the ideal modular case with SMI's of 0.87, while the cell phone and laptop tend towards the more integral "bus-modular" architecture of Fig.2b with SMI's of 0.78 and 0.83, respectively. Finally, the singular value decay curves of Fig.9 also provide support for our hypothesis. The curve decays quicker for the two products with technical performance constraints (cellular phone and laptop) than for the two business factor driven products (desk phone and desktop PC).

As discussed above, all three metrics provide support for the hypothesis (step 6). Therefore we accept the hypotheses and conclude that technical constraint based design of engineering systems leads to a higher degree of coupling between elements of form and functional elements (= more integral architectures) relative to systems were such constraints are inactive.

8 Discussion

We found support for our hypothesis that technical constraint based design of engineering systems leads to more integral architectures. This was supported by all three metrics we used. The result is significant, since the three metrics all measure modularity from a different perspective. Nonetheless, in all three cases, we noticed that the product with technical constraints is more integral in terms of more functions per components and in terms of connectivity both within and between modules. The degree of modularity for those products, however, is closer to the ideal bus-modular case of Fig.2b, than it is to the fully integral case of Fig.2a.

We observed that the interaction profiles in the DSM matrix of performance critical and non-performance critical systems are significantly different. We find that the higher the performance requirement or constraint of a system the higher the coupling between system elements. Also, the weight and power constrained laptop computer and cellular phone are more coupled than the desktop PC and desk phone respectively. We showed this through examples with two different product pairs and by using three different metrics to measure modularity and coupling in design. The results are significant since all three metrics measure modularity or coupling from a different viewpoint, but all three results in the same conclusion.

The result is in apparent violation of the independence axiom in Axiomatic Design as well as the idea of modularity being always a virtue. We show that, at least in certain cases, the axiomatic or modular design has to be compromised by technical concerns such as efficiency of weight, packaging and/or power. The insight is that in performance critical design, considerations other than pure axiomatic or modular design must be considered. But as important, is to note that modular and axiomatic design has its place in design when business drivers are more important than the technical drivers. In addition we notice that the integrality of the architectures results in more connected product structures that require more effort to redesign if the design requirements were to change.

Future work will potentially include dissection and analysis of a larger array of products and non-dimensional analysis to quantify scaling laws between modularity/integrality and performance/efficiency across a variety of domains.

9 Conclusions

We have shown how modular design is not the ideal solution in designing technical constraint driven systems. If technical constraints, such as power consumption or weight, are the main drivers of design, an integral system will provide a more suitable architecture than a modular system. A uncoupled modular design, no the other hand, is preferred, when business drivers, such as commonality and flexible design, are the main concern during design. We showed this through examples with 3 different systems and by using 3 different metrics to measure modularity and coupling in design. The results are significant since all three metrics measure modularity or coupling from a different viewpoint, but all three result in the same conclusion.

References

[1] Aarnio J., “Modularization by Integration: Creating Modular Concepts for Mechatronic Products”, PhD Thesis, Tampere University of Technology, 2003.

[2] Baldwin C. Y. and Clark K. B., “Design Rules: Volume 1. The Power of Modularity”, The MIT Press, Cambridge, MA, 2000.

[3] Bass, L., Clements, P., and Kazman, R., “Software architecture in practice”, 2nd ed. Addison-Wesley, 2003.

[4] Benini L. and de Micheli G., “System-level power optimization: Techniques and tools,” Proceedings of the Int. Symp. Low-Power electronics Design, San Diego, CA, 1999, pp. 288–293.

[5] Braha, D., “Partitioning tasks to product development teams”, Proceedings of the ASME Design Engineering Technical Conferences, Montreal, Canada, 2002.

[6] Cutherell D., “Product architecture”, In: The PDMA handbook of new product development, Rosenau M., Griffin A., Castellion G., and Anschuetz N. (eds), John Wiley & sons, 1996.

[7] Dong, Q. and Whitney, D., "Designing a Requirement Driven Product Development Process," Proceedings of the ASME Design Engineering Technical Conferences, Pittsburgh, PA, 2001.

[8] Eppinger S. D., "Innovation at the Speed of Information", Harvard Business Review, vol. 79(1), 2001, pp. 149-158.

[9] Ericsson A. and Erixon G., “Controlling Design Variants”, ASME Press, 1999.

[10] Fixson S. K. and Clark J. P., “On the link between modularity and cost – a methodology to assess cost implications of product architecture differences”. Proceedings of the IEEE International Engineering Management Conference (IEMC 2002). Cambridge, UK, 2002, pp. 131-136.

[11] Gonzalez-Zugasti, J.P. and Otto, K.N., “Platform-based spacecraft design: A formulation and implementation procedure”, Proceedings of the IEEE Aerospace Conference, Volume 1, 2000, p 455-463.

[12] Guo, F. and Gershenson, J. K., “A comparison of modular product design methods on improvement and iteration”, Proceedings of the ASME Design Engineering Technical Conferences, Salt Lake City, UT, 2004.

[13] Holtta K. and Salonen M., “Comparing three modularity methods”, Proceedings of the ASME design engineering technical conferences, Chicago, IL, 2003.

[14] Kazman R., Barbacci M., Klein M., Carrière S. J., and Woods S. G., “Experience with performing architecture tradeoff analysis”, Proceedings of the 1999 International Conference on Software Engineering, Los Angeles, CA, 1999.

[15] Kusiak A and Huang C., “Development of modular products”, IEEE Transactions on components, packaging, and manufacturing technology, Part A, Volume 19 (4), 1996, pp. 523-538.

[16] Kusiak A., “Engineering design: Products, Processes, and Systems”, Academic Press, 1999.

[17] Liebeck R.H., “Design of the Blended Wing Body Subsonic Transport”, Journal of Aircraft, Vol. 41, No.1, Jan-Feb 2004, pp. 10-25.

[18] Newcomb, P. J., Bras, B., and Rosen, D. W., “Implications of modularity on product design for the life cycle”, Journal of Mechanical Design, Volume 120, 1998, pp. 483-490.

[19] Pimmler T. U. and Eppinger S. D., “Integration analysis of product decompositions”, Proceedings of the ASME Design Engineering Technical Conferences, 1994, Pp. 343 – 351.

[20] Sosa M. E., Eppinger S. D., and Rowles C. M., “Identifying modular and integrative systems and their impact on design team interactions”, Journal of Mechanical Design, Volume 125, 2003, pp. 240/251.

[21] Stone, R. B., Wood, K. L. and Crawford, R. H., “A heuristic method for Identifying Modules for Product Architecture”, Design Studies, 21(1), 2000, pp. 5-31.

[22] Suh N. P., “Axiomatic Design: Advances and Applications”, Oxford University Press, 2000.

[23] Ulrich K. T., “The role of product architecture in the manufacturing firm”, Research Policy, 24, 1995, pp. 419-441.

[24] Ulrich, K. T. and Eppinger, S. D., “Product Design and Development”, 3rd ed., McGraw-Hill, 2003.

[25] Whitney D. E., “Physical limits to modularity”, Working paper, ESD-WP-2003-01.03-ESD, Massachusetts Institute of Technology, Engineering Systems Division, 2004.

[26] Whitney D. E., “Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development”, Oxford University Press, 2004.

[27] Zamirowski, E. J. & Otto K. N., “Identifying Product Family Architecture Modularity Using Function and Variety Heuristics”, Proceedings of the ASME Design Engineering Technical Conferences, Las Vegas, NV, 1999.

Corresponding author: Katja Holtta
Massachusetts Institute of Technology
Center for Innovation in Product Development
30 Memorial Dr., E60-246, Cambridge, MA 02139
USA
Phone: +1 617 452 2998
Fax: + 1 617 258 0485
E-mail: holtta@mit.edu