1999 Society for Design and Process Science
Printed in the United States of America

MODULAR PRODUCT DESIGN : A LIFE-CYCLE VIEW

J. K. Gershenson*, G. J. Prasad **, S. Allamneni*
*Department of Mechanical and Aerospace Engineering
Utah State University
Logan,Utah
******Plumlee Associates , Inc.
Baton Rouge,Louisiana

This paper discusses the incorporation of modularization into mechanical designs. The research uses a definition of modularity that incorporates the potential of modularity based not only on the form/function structure of a product but also life-cycle processes such as manufacture, assembly, service, and recycling.

Modularization, due to the functional independence it creates, has been called the goal of good design. Industry has made an effort to modularize products to be flexible to the needs of end users. In addition, some modules are created with some aspects of assembly in mind. Life-cycle modularity entails maintaining independence between components and all life-cycle processes in different modules, encouraging similarity in all components and processes in a module, and maintaining interchangeability between modules.

In this paper, the definition of product modularity is given. A measure of relative modularity and a modular design methodology are developed that encourage modularity, prevent a cascade of product design changes due to changes in life-cycle processes, and support agile reaction to changes in life-cycle processes. A short example is used to clarify the work.

1. Introduction

Modularity is a common but unexplored thread among all areas of life-cycle engineering. Modular products tend to have fewer components for assembly and are therefore cheaper to assemble. Modularity allows for the reduction of service costs by grouping components so those less reliable components are easily accessed. In addition, grouping components into modules by how they are recycled can greatly reduce product retirement costs.

1.1. Previous research into modularity

Most research into modularity originates from Suh's (1990) independence axiom that states, "in good design the independence of functional requirements is maintained."

Therefore, if possible, each function that a product performs should be independent of all other functions the product performs. This axiom has led to a search for a connection between physical independence and functional independence. In one of the first works to discuss modular design theory, Ulrich and Tung (1991) use product modularity as a design goal. They define modularity in terms of two characteristics of product design: "1) Similarity between the physical and functional architecture of the design and 2) Minimization of incidental interactions between physical components." In an extension of this work, Ulrich (1995) states that a modular product or subassembly has "a one-to-one mapping from functional elements in the function structure to the physical components of the product" and that all interfaces between the components of different modules are decoupled.

Newcomb, *et al.* (1996) discuss the role of product architecture in modular design. They look at the effect of modular architecture on the product life-cycle. While the authors do not discuss how to use modularity to affect life-cycle cost, they do understand that modularity influences cost. While Chang and Ward (1995) use a more dynamic application of functional modules, Erixson (1996) and Kusiak (1996) detail design for modularity as a tool to decrease assembly and manufacturing costs in product families and manufacturing systems.

Chen, *et al.* (1994) propose a measure of modularity based upon the independence of functional requirements and their sensitivity to changes in design parameters. However, their work could not account for the interrelationship between the sensitivity and independence (Rosen, 1995). DiMarco, *et al.* (1994) created a software tool that assesses a qualitative product recycling cost. Their definition of modules, based on physical characteristics and designer's intent, does not capture all aspects of product modularity; neither do the authors develop a methodology for creating recycling modules.

In summary, work exists which defines modularity, develops methods of measuring modularity, and applies modularity to product design. Some aspects missing in the current works include a definition for modularity that takes into account aspects of a product other than its function, a methodology for designing modular products, and an accompanying modularity measure.

1.2. Benefits of modularity

Modularity allows the designer to control the degree to which changes in processes or requirements affect the product and by promoting interchangeability, modularity gives designers more flexibility to meet these changing processes. This flexibility allows for delaying design decisions until more information is available without delaying the product development process. Another benefit is the ability of modularity to reduce life-cycle costs by reducing the number of processes and reducing repetitive processes.

Ulrich and Tung's (1991) work details the costs and benefits of modular products. The benefits of modularity they discuss include 1) component economies of scale, 2) ease of product updating, 3) increased product variety, 4) decreased order lead-time, and 5) ease of design and testing. The costs of modularity they discuss include 1) static product architecture, 2) lack of performance optimization, 3) increased unit variable costs, and 4) excessive product similarity. Other works (Ishii, 1995; Shah, 1996; Ulrich, 1991) concentrate on the benefits in the design process.

2. Definitions

Life-cycle modularity is the foundation of this work. Life-cycle modularity is a relative property. Products possess a higher or lower degree of modularity. A product with a higher degree of modularity either contains a larger percentage of components or subassemblies that are modular or contains components and subassemblies, which are, on average, more modular. Subassemblies, which are relatively modular in nature, are *modules*.

Modules contain a high number of components that have minimal dependencies upon and similarities to other components not in the module. These dependencies and similarities include those that arise from the component interactions and those which arise from the various processes the components undergo during their life-cycle. In an ideal module, each component is independent of all components not contained in that module throughout the entire product life-cycle (independence). In addition, each component in the module is processed in the same manner during each life-cycle stage (similarity) (Gershenson and Prasad, 1997). This definition expands the form-function relationship to a form-process relationship. Similarity is a new perspective on the separation of form and process. Each part of the form (module) must undergo the same life-cycle processes. Independence and similarity represent a significant increase in the rigor of defining product modules versus past form/function independence.

An important consideration when defining the relative life-cycle modularity of a product is the level of detail chosen when looking at the product structure. A product may seem modular but, at some levels of detail, the structure may not be modular. We use *component trees* as a tool to describe the levels of detail of a product. Component trees show all of the components and subassemblies that make up the product. Components can be examined down to their constitutive attributes (material, geometry, tolerances, features, *etc*.) (Gershenson and Stauffer, 1995). The tree-like structure is helpful in discerning levels of detail and showing subassembly interactions.

Another important consideration when defining life-cycle modularity is the chosen level of abstraction of the life-cycle process itself. As an example, product retirement consists of many tasks (*e.g*., reuse, remanufacture, recycle). These tasks are made up of subtasks (*e.g*., collection, separation, and grinding.). A product may seem modular when examined from the standpoint of the overall retirement process but at some subtask level, the product may not be modular. *Process graphs* are used to describe the levels of detail of life-cycle processes. Process graphs delineate each task and subtask of a process.

Creating modular products involves comparing the component tree and process graphs of a product and making sure that, at each level of detail, the product's attributes are as independent from one another as possible for each level of detail of the life-cycle processes. If a dependency does occur, it should occur within a module. In addition, within a module and at each level of detail, every process should be similar for every component. Lastly, depending upon the product, the connections between the modules should allow for the interchangeability of modules.

To increase independence and similarity, a product must be designed with the following three facets of modularity:

Attribute Independence: Component attributes have fewer dependencies on attributes of other modules, called external attributes. If there are dependencies, fewer attributes are dependent upon one another and attributes that are related to external attributes are less dependent. *E.g.*, Lego® pieces which can be of any color, size, shape, or material as long as they have the correct dot to attach to other pieces and an impression to accept other pieces. Attribute independence allows for the redesign of a module with minimized effects on the rest of the product. Attribute similarity is excluded because having similar but unrelated components is not detrimental as long as attribute independence is maintained.

Process Independence: Each task of each life-cycle process of each component in a module has fewer dependencies on the processes of external components. This requires that the processes a module undergoes during its life-cycle are independent of the processes undergone by external modules. Any dependencies that do exist are minimized in number and criticality. *E.g.*, in separation for recycling, techniques that utilize grinding and separation by material density are dependent upon the disassembly of all components containing materials that are not compatible and are of a similar density. If the disassembly process occurred later in the retirement process, grinding and density separation would not be possible. Process

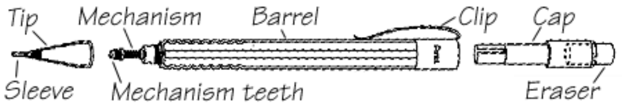


Fig. 1 Exploded view of the mechanical pencil highlighting the four modules: cone/tip, clutch/ teeth, barrel, and eraser.

independence allows for the reduced cost in each life-cycle process and the redesign of a module in isolation if processes should change.

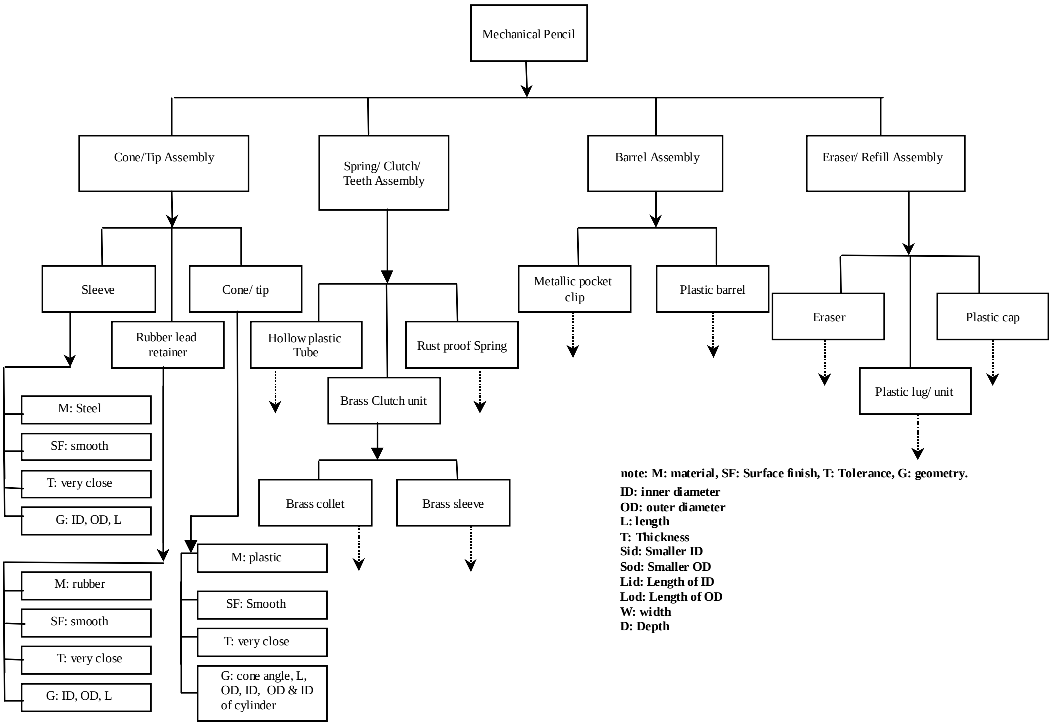
Process Similarity: Group components and subassemblies that undergo the same or compatible life-cycle processes into the same module. *E.g.*, if a product is being recycled through grinding, it would be least expensive if all components undergoing this task were in the same module therefore the entire module could be ground and then no other grinding would be necessary. Process similarity minimizes the number of external components that undergo the same processes, creates a strong differentiation between modules, reduces process repetition, and reduces process costs. Process similarity also conserves redesign effort by insuring that changes to individual life-cycle processes only affect one module of the product.

3. Modularity Measure

The modularity measure, like most measures of “goodness” or “X-ability,” is best used in comparing the relative modularity of two like products. There is quite a bit of initial work in calculating the measure. Analysis with only a few life-cycle processes in life-cycle modularity may be more useful than total product modularity for a product, especially if a particular life-cycle process dominates the requirements of a product. The four-step measure that follows relies heavily on understanding the physical and process relationships among components. The example application used throughout the next two sections of the paper is the mechanical pencil in Figure 1.

Step 1: Generating a Component Tree - A component tree details the physical relationships among components at all levels of abstraction. To develop a component tree, the product is divided into its constitutive modules and components. The modules are further classified into subassemblies, then individual components, and lastly product attributes that describe the components. A partial component graph for a mechanical pencil highlighting the attributes of the cone/tip assembly is shown in Figure 2. From Figure 2, it can be seen that the cone/tip assembly is comprised of components such as sleeve, rubber lead retainer, and cone/tip with similar geometric attributes (ID and OD for each) but very different material attributes (steel, rubber, and plastic).

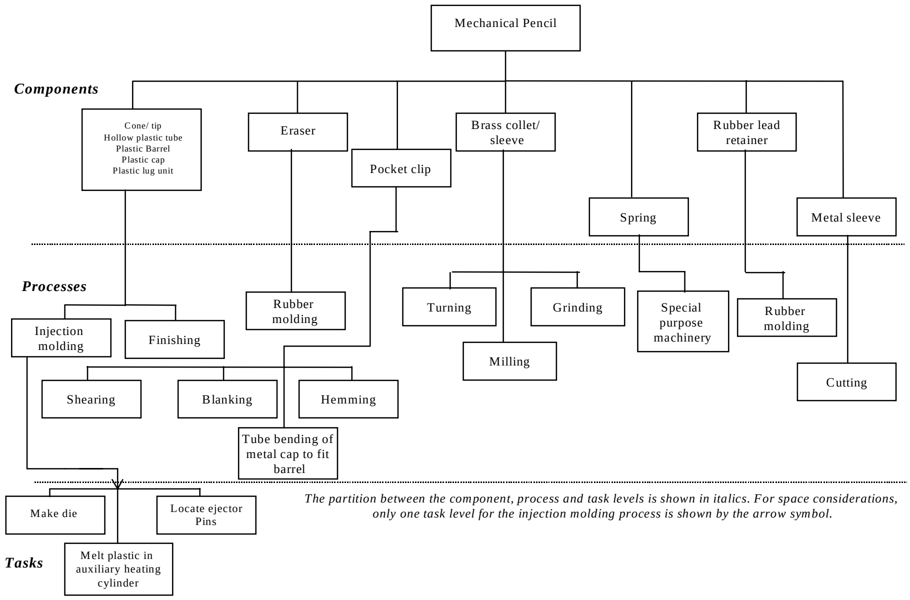
Step 2: Generating Process Graphs - The various life-cycle processes that each of the components in all of the modules undergo are first jotted down and then all the components which undergo each life-cycle process (e.g., manufacturing, assembly, function, service, retirement) are noted. For each process, a process graph must be created that details each stage of the life-cycle, all of the processes in each stage, and each of the tasks and subtasks in each process. The manufacturing graph for a mechanical pencil is shown in Figure 3. Components are grouped together according to the manufacturing process undergone and then each manufacturing process is expanded to include the pertinent tasks and subtasks of each process. The components cone/tip, hollow plastic tube, plastic cap, and plastic lug unit are all plastic



note: M: material, SF: Surface finish, T: Tolerance, G: geometry.

ID: inner diameter
OD: outer diameter
L: length
T: Thickness
Sid: Smaller ID
Sod: Smaller OD
Lid: Length of ID
Lod: Length of OD
W: width
D: Depth

Fig 2. A partial component tree of a mechanical pencil cone/tip assembly.



The partition between the component, process and task levels is shown in italics. For space considerations, only one task level for the injection molding process is shown by the arrow symbol.

Fig 3. A partial manufacturing process graph for the mechanical pencil example.

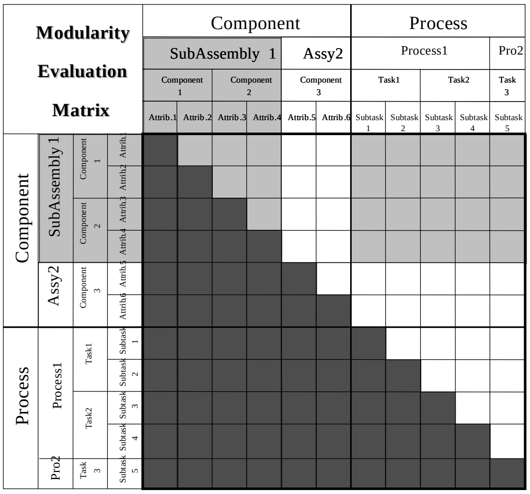


Fig 4. A generalized modularity evaluation matrix. Each subassembly and process is broken down into its constitutive elements, attributes, and subtasks. The boxes contain the weights of the similarity and dependency relationships.

Similarity	Dependency
1: Not similar	1: Not dependent
2: Slightly similar	3: Dependent
3: Similar	5: Highly dependent
4: Very similar	
5: Extremely similar	

Fig 5. Similarity and dependency ratings.

injection molded components. The injection molding process is further classified into the tasks making the die, locating the ejector pins, and melting the plastic in the auxiliary heating cylinder.

Step 3: Construction of the Matrices - Using the component tree and process graphs, two modularity evaluation matrices are constructed, one to record similarities and one to record dependencies. Figure 4 shows the general form of the modularity evaluation matrix. The square matrix has row and column headings corresponding to the most specific levels of the component tree and process graphs. The contents of the two modularity evaluation matrices, are the similarity and dependency relationships among components and processes.

There are six possible relationships within both similarity and dependency:

Component-Component Dependency occurs when two components are reliant upon each other with respect to their physical design, specifically their attributes. An example of this is a gear that fits on a shaft. While the gear and the shaft are two different components, the inner diameter of the gear and the outer diameter of the shaft are inextricably dependent upon each other.

Component-Component Similarity is not used because it does not tie the designs together so as to necessitate changes in one due to changes in the other.

Component-Process Dependency details relationships in which product design is contingent upon the life-cycle process a component undergoes, *i.e.*, process drives design. If the same process drives the designs of two different components, the components should be grouped in the same module so that they can evolve with the process and minimize effects on other components. One simple example is a tuner dial and a power switch on a stereo, the two components are totally unrelated but undergo the same manufacturing process. All such plastic injection molded components could be combined into one module so that they can be updated as one with changes in the injection molding process.

Component-Process Similarity details relationships in which a component uses or goes through the life-cycle process. The logic is to group components that undergo the same life-cycle processes in one module to minimize the impact a change in process will have on the product. As an example two fiberglass components of a motorcycle such as the front and back mudguard which are manufactured and retired by the same process or for that matter any other fiber glass component, which is assembled in the same stage of assembly. These components can be placed in the same module irrespective of their locations.

Process-Process Dependency and *Process-Process Similarity* do not affect product design directly, due to the exclusion of component interaction, and have been excluded from both the relative modularity measure and design methodology.

A set of ratings, the contents of the modularity evaluation matrices, is shown in Figure 5. As an example, referring to Figure 6, a rating of 1 is given for the component-component similarity between the rust proof spring and the sleeve whereas a rating of 5 is given for the component-component similarity between the brass collet and the brass sleeve. This differential is due mostly to a similarity in material attributes between the collet and sleeve. In addition, a rating of 5 is given to the component-process similarity between the component cone/tip and the injection molding process because the entire cone/tip is injection molded.

Step 4: Calculation of the Relative Modularity using the Modularity Evaluation Matrix - For a high degree of modularity, it is important to have a high similarity between components within a module (S_{in}), a low similarity between a component of a concerned module and other components outside of the module (S_{out}), a high dependency between components within the module (D_{in}), and a low dependency between a component within a module and a component outside of the module (D_{out}).

The measure of relative modularity that was finally developed is:

$$\text{Modularity} = S_{in} / (S_{in} + S_{out}) + D_{in} / (D_{in} + D_{out})$$

The above measure directly correlates to the definitions of similarity and dependence put forth in the previous sections. In this measure we find the ratio of the sum of similarities inside the modules to the total similarity and sum it up with the ratio of the sum of dependencies inside the modules to the total dependency. The values that are obtained for the whole product show increasing modularity from 0 to 2.

A brief description of the calculation of the four prime parameters (S_{in}, S_{out}, D_{in}, D_{out}) is given below:

S_{in}: <u>Component similarities between each component within a particular module.</u>

$$S_{in} = \sum_{m=1}^{M} \sum_{i=r}^{s-1} \sum_{j=i+1}^{s} \sum_{k=1}^{T} \sqrt{S_{ik} * S_{jk}}$$

Similarity and depend			Tree	CG	CG	CG	CG	CG	CG	CG	CG	CG	CG	CG	CG	CG
evaluation matrix			Assy	cone/tip	cone/tip	cone/tip	Teeth/clt	Teeth/clt	Teeth/clt	Teeth/clt	Teeth/clt	Barrel	Barrel	Eraser	Eraser	Eraser
			Comp/Assy	Sleeve	Cone	RL Ret.	Pl. Tube	RP Spring	Clutch Un	Clutch Un	Clutch Un	M Clip	Pl Brl.	Eraser	Pl. Cap	Pl. lug
Tree	Assy	Comp/Ass	Comp						Br. Collet	Br. Sleeve	Collet Cyl.					
CG	cone/tip	Sleeve		0/0	3/5	2/5	3/0	1/0	5/3	5/3	2/0	3/0	2/3	2/0	3/0	4/0
CG	cone/tip	Cone		3/5	0/0	3/5	3/0	1/3	1/5	2/0	2/0	2/0	3/5	3/0	3/0	3/0
CG	cone/tip	RL Ret.		2/5	3/5	0/0	2/0	1/3	2/3	2/0	2/0	2/0	4/0	3/0	3/0	3/0
CG	Teeth/clt	H Pl. Tube		3/0	3/0	2/0	0/0	2/3	4/3	4/5	4/5	2/0	3/3	2/0	3/0	4/5
CG	Teeth/clt	RP Spring		1/0	1/3	1/3	2/3	0/0	4/0	4/5	4/5	5/0	4/2	4/0	4/0	4/0
CG	Teeth/clt	Clutch Un.	Br. Collet	5/3	1/5	2/3	4/3	4/0	0/0	5/5	5/5	5/0	2/0	1/0	2/0	2/0
CG	Teeth/clt	Clutch Un.	Br. Sleeve	5/3	2/0	2/0	4/5	4/5	5/5	0/0	5/5	5/0	2/2	4/0	2/0	2/0
CG	Teeth/clt	Clutch Un.	Collet Cyl.	2/0	2/0	2/0	4/5	4/5	5/5	5/5	0/0	5/0	2/3	1/0	2/0	2/0
CG	Barrel	M Clip		3/0	2/0	2/0	2/0	5/0	5/0	5/0	5/0	0/0	3/5	2/0	2/0	2/0
CG	Barrel	Pl Brl.		2/3	3/5	4/0	3/3	4/2	2/0	2/2	2/3	3/5	0/0	3/0	3/0	3/0
CG	Eraser	Eraser		2/0	3/0	3/0	2/0	4/0	1/0	4/0	1/0	2/0	3/0	0/0	4/5	4/5
CG	Eraser	Pl. Cap		3/0	3/0	3/0	3/0	4/0	2/0	2/0	2/0	2/0	3/0	4/5	0/0	5/5
CG	Eraser	Pl. lug		4/0	3/0	3/0	4/5	4/0	2/0	2/0	2/0	2/0	3/0	4/5	5/5	0/0

				MG	MG	MG	MG	MG	MG	MG	MG	MG	MG	MG
				Cone	Era. Uni	Pkt. Clip	Pkt. Clip	Pkt. Clip	Pkt. Clip	Collet	Collet	Collet	Spring	Metal Slv.
				Inj. Mold	Rub. Mol	Shearing	Blanking	ube bend	Hemming	Turning	Grinding	Milling	SPM	Cutting
Tree	Assy	Comp/Ass	Comp	.	omp. Mol	Sh. Punc	nk punch	e punch &	e punch &	me, colle	job temp	milling c	re drawin	e cutting
CG	cone/tip	Sleeve		0/0	0/0	0/0	0/0	0/0	0/0	5/5	5/5	5/5	0/0	5/5
CG	cone/tip	Cone		5/5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
CG	cone/tip	RL Ret.		0/0	5/5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
CG	Teeth/clt	H Pl. Tube		5/5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
CG	Teeth/clt	RP Spring		0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	5/5	0/0
CG	Teeth/clt	Clutch Un.	Br. Collet	0/0	0/0	0/0	0/0	0/0	0/0	5/5	5/5	5/5	0/0	0/0
CG	Teeth/clt	Clutch Un.	Br. Sleeve	0/0	0/0	0/0	0/0	0/0	0/0	5/5	5/5	5/5	0/0	0/0
CG	Teeth/clt	Clutch Un.	Collet Cyl.	0/0	0/0	0/0	0/0	0/0	0/0	5/5	5/5	5/5	0/0	0/0
CG	Barrel	M Clip		0/0	0/0	5/5	5/5	5/5	5/5	0/0	0/0	0/0	0/0	0/0
CG	Barrel	Pl Brl.		5/5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
CG	Eraser	Eraser		0/0	5/5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
CG	Eraser	Pl. Cap		5/5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	5/5
CG	Eraser	Pl. lug		5/5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0

Fig 6. Partial similarity and dependence modularity evaluation matrix for the mechanical pencil. The first value of each cell represents similarity; the second value represents dependence. All process rows and the attribute levels of components are eliminated to save space. The columns represent the elements of the component and the process graph.

Where: m *is a module,* i, j *are components in the same module, and* k *is a task*
M = *# of modules in the product*
r = *first component in module* m *or module* n
s = *last component in the module* m *or module* n
T = *# of processes under consideration*
S_{ik} *is similarity between component* i *and task* k
S_{ik} *is similarity between component* j *and task* k

This value is the root mean square of the similarities between two components and a life-cycle process. Like all of the component-process measures to follow, it allows a component-process relationship to be measured in component-component terms. S_{in} is calculated for only component-process interaction. Hence the calculation of S_{in} for a component of module A makes use of the ratings of the component-process similarity interactions for each component *within module A.* S_{in} has a positive effect on the measure as we are trying to group components with similar life-cycles.

S_{out}: <u>Similarities between the components of a module and each component external to the module.</u>

$$S_{out} = \sum_{m=1}^{M}\sum_{i=r}^{s-1}\sum_{n=m+1}^{M}\sum_{j=r}^{s}\sum_{k=1}^{T}\sqrt{S_{ik} * S_{jk}}$$

Where i, j are components not in the same module, and n is a module.

The calculation of S_{out} for a component of module A makes use of the ratings of the component-process similarity interactions for each component *outside module A.* S_{out} has a negative effect on the modularity measure, as we are interested in reducing process similarities between components that are in two different modules.

D_{in}: <u>Dependencies between each component within a particular module.</u>

D_{in} = Component-Process interactions + Component-Component interactions

$$D_{in} = \sum_{m=1}^{M}\sum_{i=r}^{s-1}\sum_{j=i+1}^{s}\sum_{k=1}^{T}\left(\sqrt{D_{ik} * D_{jk}} + D_{ij}\right)$$

Where: i, j *are components in the same module*
D_{ik} *is the dependence between component* i *and task* k
D_{jk} *is the dependence between component* j *and task* k
D_{ij} *is the dependence between component* i *and component* j

The calculation of D_{in} for a component of module A makes use of the ratings of the component-component *and* component-process dependence interactions for each component *within module A.* Component-component dependencies are taken right out of the modularity evaluation matrix. D_{in} has a positive effect on the measure as it is important to group dependent components.

D_{out}: <u>Dependencies between the components of a module and each of the components that are external to the module.</u>

D_{out} = Component-Process interactions + Component-Component interactions

$$D_{out} = \sum_{m=1}^{M} \sum_{i=r}^{s-1} \sum_{n=m+1}^{M} \sum_{j=r}^{s} \sum_{k=1}^{T} \left(\sqrt{D_{ik} * D_{jk}} + D_{ij} \right)$$

Where: *i, j are components not in the same module*
D_{ik} *is the dependence between component i and task k*
D_{jk} *is the dependence between component j and task k*
D_{ij} *is the dependence between component i and component j*

The calculation of D_{out} for a component of module A makes use of the ratings of the component-component *and* component-process dependence interactions for each component *outside module A.* D_{out} has a negative impact on the total measure, as all external dependencies must be minimized to have independent modules.

Cursory Example of a Mechanical Pencil

In this brief example, we have calculated the modularity of the ***cone/tip assembly only*** taking into account only component, function, and manufacturing interactions. The relationships are shown in Figure 6 at the end of the paper. The first step in measuring the modularity of the mechanical pencil is to calculate the values of S_{in}, S_{out}, D_{in}, and D_{out}.

$S_{in} = 0$

If we refer to the matrix in Figure 6, we see at the bottom part of the matrix for the cone/tip assembly that none of the processes are common and hence there are no similarities.

$$S_{out} = S_{\text{injection molding}} + S_{\text{rubber molding}} + S_{\text{function 1}} + S_{\text{function 2}}$$

$$= (\surd(5*5)+\surd(5*5)+\surd(5*5)+\surd(5*5)) + (\surd(5*5)) + (3*(\surd 5*5 + (\surd 5*5) + (\surd 5*5)) + (\surd(5*5)) +(\surd(5*5))$$

$\Rightarrow \quad S_{out} = 80$

In the manufacturing section of the matrix, there are external similarities in the processes injection molding and rubber molding and for the two functions lead slide and stopping of lead slide (not shown in the matrix). The value for S_{out} for each of the above processes, for only the cone/tip assembly, is calculated by applying the formula for S_{out} from the previous section and summing up the similarities for all the process. As an example, in injection molding there are four external components that undergo the same process. Therefore, there are four sets of $\surd(5*5)$ at the beginning of the calculation of S_{out}.

$D_{in}=D_{in,c-c}+D_{in,c-p}$

$D_{in,c-c} = 10+5=15$ $\qquad$ $D_{in,c-p} = 0$

$\Rightarrow D_{in}= 15$

D_{in} includes both the component-component dependencies and component-process dependencies inside the module. Component-component dependencies are depicted in the top part of the matrix whereas component-process dependencies are computed from the bottom part of the matrix. The method for calculating both the dependencies is explained in the previous section. $D_{in,c-c}$ for the cone/tip assembly consists of two 5-weight dependencies between two components, the sleeve and the cone and the sleeve

Table 1. Total relative modularity of the mechanical pencil and each of its four modules.

Module	S_{in}	S_{out}	D_{in}	D_{out}	RM
Cone/Tip	0	80	15	98	0.13
Teeth/Clutch	45	75	131	101	0.94
Barrel	0	20	5	39	0.11
Eraser	5	35	20	40	0.46
Total Relative Modularity				**0.87**	

RM = modularity of the component; TRM $\neq$ SRM

and the retainer. There is also a 5 weight relationship between the retainer and the cone. This leads to a 10 as the first part of $D_{in,c-c}$ and a 5 as the second. There are no common processes so $D_{in,c-p} = 0$.

$D_{out}=D_{out,c-c}+D_{out,c-p}$

$D_{out,c-c} = 9+13+6 = 28 \qquad D_{out,c-p} = 20+5+45 = 70$

Þ $D_{out} = 98$

$D_{out,c-c}$ involves outside dependencies between, for example, the sleeve and the brass collet, the brass sleeve, and the plastic barrel. Each dependency has a weight of 3 leaving a 9 as the first value in $D_{out,c-c}$.

Once the calculations for S_{in}, S_{out}, D_{in}, and D_{out} are made for all four modules of the mechanical pencil, the relative measure for the product can be calculated by summing up the relative measures of each of the modules as in Table 1. The table displays the relative modularity values for each of the four modules and the relative measure for the mechanical pencil as a whole. Notice that the values in the cone/tip row coincide with those described above. The cone/tip scored low because it consists of varied components unlike the teeth/clutch, which has components that are more congruous. The relative modularity for the pencil is 0.87 where the possible range of values is 0 to 2; the pencil scored low. This value has little meaning by itself but is useful to compare design options and to guide the redesign process.

4. Modular Design Methodology

Using the definition of characteristic modularity, a specific methodology for designing products that are modular in terms of their life-cycle processes was developed. The design methodology is a set of quantitative guidelines that direct product development towards modular products with all of the benefits therein.

The goal of the design methodology is to redesign a product eliminating components or modules, rearranging components or modules, or changing component attributes. Elimination is the simplest process. Reconfiguration is the cost effective shifting of components to other modules to increase the total relative modularity. Redesign is the changing of the component attributes to reduce outside similarities and dependencies or increase inside similarities and dependencies.

Redesign is more difficult than reconfiguration because there is a need to redo the engineering analysis. The logic of the design methodology, is as follows:

1. eliminate the modules if they are not necessary;
2. if the whole module cannot be eliminated, then look to eliminate the components of these modules;

Table 2. Total relative modularity of the mechanical pencil after application of the modular design methodology.

Module	**S_{in}**	**S_{out}**	**D_{in}**	**D_{out}**	**RM**
Cone/Tip	28	116	10	95	0.13
Teeth/Clutch	45	75	131	109	0.94
Eraser	5	40	20	43	0.46
Total Relative Modularity				**0.94**	

RM = modularity of the component; TRM $\neq$ SRM

3. if elimination is impossible, then try to shift the components to other modules or into new modules to increase the overall value of product modularity;
4. if reconfiguration is not possible, redesign the attributes of the components to decrease or eliminate similarities or dependencies with outside components or increase similarities with components of the same module.

The complete algorithm for modular product design is not included due to its length. In the algorithm, the module with the lowest of the relative modularity is first taken up for analysis as long as its modularity has changed since it was last redesigned (in an iterative effort). The relative modularity of all components in that module is calculated and the component with the lowest relative modularity is approached first. If the component cannot be eliminated, it is then taken up for reconfiguration. For reconfiguration, it is necessary to determine into which other module the component can be moved. Once the component has been shifted to another module, a feasibility check is carried out to determine the practicality of shifting. If reconfiguration is impossible, the component attributes are redesigned. This cycle is continued until the components are shifted to the modules that yield the highest total relative modularity.

Cursory Example of a Mechanical Pencil

The redesign methodology was applied for three "rounds" to the mechanical pencil. There were significant improvements in the value of the total relative modularity. First, we calculated the relative modularities shown in Table 1, leading us to approach the barrel assembly first. Since elimination of the barrel module or any other module was impossible, the next step was to approach, in order of relative modularity, the components of the modules. Again, no opportunity for elimination occurred. The next step is to look for possible opportunities for reconfiguration. While there were opportunities to increase modularity, they did not yield feasible products. The last step was to attempt to redesign a component. Beginning with the lowest module, the barrel assembly, we approached the components in order of worst RM. The plastic barrel was the first candidate but nothing could be done. Next came the metallic clip, we redesigned the clip's material attribute from metal to plastic. This increased the modules relative modularity due to increased process similarity within the module. Then, it was back to the beginning of the methodology and, this time only searching those modules and components that had experienced a change in the first round. After a recalculation of the module and component RMs, we again came all the way down to redesign before finding a possibility. Again, the barrel assembly was worst with the plastic barrel component being worst in the module. Now that they were of the same material, the plastic barrel and the clip could be combined in a single component. Again, the relative modularity of the modules and components were recalculated and the barrel assembly was *still* lowest. After elimination was ruled out, reconfiguration was attempted. By comparing [S_{inj} + D_{inj}] with [S_{outjk} + D_{outjk}] for each module, it was determined that the plastic barrel (with integral clip) could be reconfigured and, based on similar processes and that the threaded portion of the barrel is assembled to the inner diameter of the cone/tip, the component plastic barrel/clip was moved to the cone/tip assembly. This eliminated the barrel assembly since there were no components left in that module. Note that for elimination and reconfiguration steps, some additional redesign is usually necessary. The value of total relative modularity was increased from 0.87 to 0.94. The new values of relative modularity are described in Table 2.

5. Conclusion and Future Work

In this paper, we have discussed the development of a set of definitions that structure life-cycle product modularity. We have also proposed a relative modularity measure and design methodology. By using a structured modularity measure built upon component independence, process similarity, and process independence, it is possible to compare the degree to which a design is enjoying the benefits of modularity. Our approach focuses on independence and similarity across the life-cycle and includes a step-wise redesign methodology to guide designers towards modular products. It is important to view product modularity from the standpoint of creating more modular products. This is quite different from designing products with interchangeable or reconfigurable parts. It is also quite different from maintaining form/function independence. It is the goal of modular design to group all attributes with like life-cycle processes into a single module and decouple them from all other attributes and life-cycle processes.

The example product chosen for our analysis is a mechanical pencil. We have deliberately chosen a simple product. Our intention is to show the measure and apply the methodology before the final phase of testing on several comparison products and a more complex product. In the example, we were able to use the measure and methodology to significantly increase the modularity of the product. Through our cursory example, we have seen that one shortcoming of the modular design methodology is the work necessary to apply it. The matrices necessitate deep product knowledge and tedious work. We are working to automate the evaluation and reconfiguration.

Our next task will be to explore the ties between modularity and life-cycle cost. Using the relative modularity measure and life-cycle cost calculations; we will research the point at which the wastes of redundancy and additional features outweigh the benefits of more efficient product families and flexibility in meeting all customer needs. In addition, we are moving towards improved development on the characteristics of similarity and dependence in each life-cycle area. This will lead to a better measure of the similarity and dependence relationships between components.

6. Acknowledgments

The authors gratefully acknowledge The University of Alabama Research Grants Committee, Alabama EPSCoR - NSF, and the National Science Foundation/Lucent Technologies Industrial Ecology Fellowship program.

7. References

Chang, T.S. and A.C. Ward (1995), “Design-in-Modularity with Conceptual Robustness,” *Proceedings of the 1995 ASME Design Technical Conferences - 21st International Conference on Advances in Design Automation*, Boston, Massachusetts.

Chen, W., *et al.* (1994), “Modularity and the Independence of Functional Requirements in Designing Complex Systems,” *Proceedings of the 1994 ASME Design Technical Conferences - Conference on Concurrent Product Design*, Minneapolis, Minnesota.

DiMarco, P., C.F. Eubanks, and K. Ishii (1994), “Compatibility Analysis of Product Design for Recyclability,” *Proceedings of the 1994 ASME Design Technical Conferences - 14th International Conference on Computers in Engineering*, Minneapolis, Minnesota.

Gershenson, J.A. and L. Stauffer (1995), “The Creation of a Taxonomy for Manufacturability Design Requirements,” *Proceedings of the 1995 ASME Design Technical Conferences - 7th International Conference on Design Theory and Methodology*, September, 1995, Boston, Massachusetts.

Gershenson, J.K. and G.J. Prasad (1997), “Modularity in Product Design for Manufacturing,” *International Journal of Agile Manufacturing*, Volume 1, Issue 1, Society of Agile Manufacturing, August, 1997.

He, D.W. and A. Kusiak (1996), “Performance Analysis of Modular Products,” *International Journal of Product Research*, Vol. 34, No. 1, pp. 253-272.

Ishii, K., C. Juengel, and J.F. Eubanks (1995), “Design for Product Variety: Key to Product Line Structuring,” *Proceedings of the 1995 ASME Design Technical Conferences - 9th International Conference on Design Theory and Methodology*, Boston, Massachusetts.

Newcomb, P.J., B. Bras, and D.W. Rosen (1996), “Implications of Modularity on Product Design for the Life-cycle,” *Proceedings of the 1996 ASME Design Technical Conferences - 10th International Conference on Design Theory and Methodology*, Irvine, California.

Rosen, D. (1995), personal conversation at ASME Design Technical Conferences, September, 1995.

Shah, *et al*. (1996), “Research Opportunities in Engineering Design,” *NSF Strategic Planning Workshop Final Report*.

Suh, N.P. (1990), *The Principles of Design*, Oxford University Press.

Ulrich, K. (1995), “The Role Of Product Architecture In The Manufacturing Firm,” *Research Policy* 24, Elsevier Science B.V.

Ulrich, K. and K. Tung (1991), “Fundamentals of Product Modularity,” *Proceedings of the 1991 ASME Design Technical Conferences - Conference on Design Manufacture/Integration*, Miami, Florida.