Proc. Natl. Sci. Counc. ROC(A)
Vol. 24, No. 3, 2000. pp. 149-165

(Invited Review Paper)

Overview of Modular Product Development

CHUN-CHE HUANG

Laboratory of Intelligent Systems and Information Management
Department of Information Management
National Chi-Nan University
Puli, Taiwan, R.O.C.

(Received December 21, 1998; Accepted August 10, 1999)

ABSTRACT

The manufacturing industry is undergoing a major paradigm shift that is taking it from traditional manufacturing into a world of agile manufacturing. An agile corporation should be able to rationalize its manufacturing facilities and produce a large variety of products at lower cost in time. Designs of modular products and reconfigurable processes are crucial to agile manufacturing. *Modular products* refer to products, assemblies and components that fulfill various functions through the combination of distinct building blocks (modules). Modular product design is becoming a focus of attention and is frequently stated as a goal of good design practice in current engineering areas. However, it has not received sufficient attention in the literature. The goal of this paper is to present the concept of modularity, review the literature on modular product design, and formulate research issues related to the development of modular products.

Key Words: modular products, product development, concurrent engineering, agile manufacturing, modules, modularity

I. Introduction

The manufacturing industry is undergoing a major paradigm shift that is taking it from traditional manufacturing into a world of agile manufacturing. An agile corporation should be able to rapidly respond to all changes in the market environment. The need for corporations to be able to deliver high quality products at low cost has long been recognized. What is also becoming clear is that the further requirements of high variety and rapid product development are gradually being superimposed on these older requirements, so that, for example:

> The complex product markets of the twenty-first century will demand the ability to quickly and globally deliver a high variety of customized products. (Earl Hall quoted in Davidow and Malone (1992)).

Customized products will be a marketing trend in the future. However, it is a challenge of manufacturing to produce variety of products with limited resources. As corporations strive to rationalize their manufacturing facilities and to produce a large variety of products at lower cost, modularity is becoming a focus of attention (Paul and Beitz, 1988). Modular products and reconfigurable processes are crucial to agile manufacturing and provide a way to produce a variety of products that satisfy various customer requirements in time (Kidd, 1994). This modular approach promises the benefits of high volume production (that arises from producing standard modules) and at the same time, the ability to produce a wide variety of products that are customized for individual customers. Such modular product design has been stated as being a goal of good design practice in current engineering areas (Kidd, 1994). However, it has not received sufficient attention in the literature (Pahl and Beitz, 1988; Shirley, 1990; Ulrich and Tung, 1991).

The goal of this paper is to present the concept of modularity, review the literature on modular design, and formulate research issues related to the development of modular products. Section II introduces the concept behind modular products. Section III presents the design of modular systems, including product, software and manufacturing systems, and illustrates the concepts with industrial examples. The difference between modular and traditional product design is explored. Section IV surveys research issues pertinent to the development of modular products. Conclusions are presented in Section V.

II. Modular Products

A product can be thought of in functional and physical terms (Ulrich and Eppinger, 1995). The functional elements of a product are the individual operations and transformations that contribute to the overall performance of the product. The physical elements of a product are the

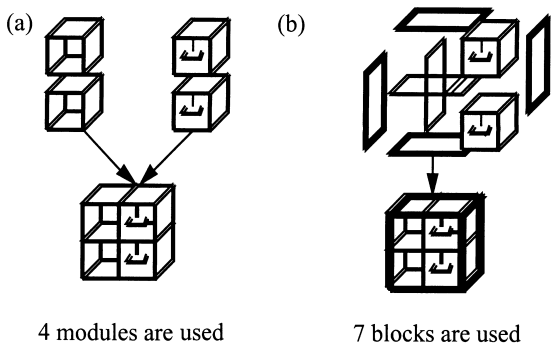

Fig. 1. Two designs of a piece of domestic furniture.

parts, components, and subassemblies that ultimately implement the product functions. The physical elements of a product are organized into several major physical building blocks. Each block is a collection of components that implement some functions of the product. A block may be a collection of interchangeable components that implement similar functions, in which case, the block is called a module. The *architecture* of a product is the scheme based on which the functional elements of the product are arranged into physical blocks and the blocks interact.

An important characteristic of a product architecture is its modularity. Consider two different designs of a piece of domestic furniture shown in Fig. 1. In the design shown in Fig. 1(a), two types of functions, the drawer and the open space, are allocated to separate modules, which in fact are mounted together and make up a piece of domestic furniture. The most modular architecture is the one where each functional element of the product is implemented by exactly one module, and in which there are a few well-defined interactions between the modules. Such a modular architecture allows a change to be made to one module without generally affecting other modules so that the product can function correctly. Each module may also be designed quite independently of other systems.

A traditional architecture is integrated much differently from the modular architecture discussed in this paper. The design shown in Fig. 1(b) is integrated, in this case motivated by ergonomic concerns. A product embodying an integrated architecture is often designed so as to maximize a certain performance measure; however, modifications to one component or feature may require extensive redesign of the product. Implementation of functional elements may be distributed across multiple blocks. Boundaries between the blocks may be difficult to identify or may not even exist.

An integrated architecture shares one or more of the following properties:

(1) The functional elements of the product are implemented using more than one block.
(2) A single block may implement many functional elements.
(3) The interactions between blocks are ill-defined and may be incidental to the primary function of the product.

Some of the motivators for product change are: upgrades, add-ons, adaptation, wear, consumption, use flexibility, and reuse. Modules allow changes to be made to a few isolated functional elements of a product without necessarily affecting the design of other elements. However, changing one block in an integrated product may influence many functional elements and require changes to several related blocks.

Product performance is defined as how a product implements its intended functions. Typical product performance characteristics are speed, efficiency, life, accuracy, and noise (Ulrich and Eppinger, 1995). To the extent that product performance depends on the size, shape or mass of a product, it generally can be enhanced by an inte-

Table 1. The Comparison of Modular and Integrated Architectures

Integrated architecture	Modular architecture
A collection of components that implement some functions of a product is called a block.	A collection of components that implement some functions of a product is called a module.
The functional elements of a product are implemented using more than one block.	Same as an integrated architecture.
A single block implements many functional elements.	A module implements one or a few functional elements in their entirety.
The interactions between blocks are ill-defined and may be incidental to the primary functions of the products.	The interactions between modules are well defined and are generally fundamental to the primary function of the product.
Product performance can be enhanced through an integrated architecture.	Product performance may not be enhanced by an modular architecture.
Changing a block in an integrated product may influence many functional elements and require changes to several related blocks.	Changing a few isolated functional elements of a product may not affect the design of other modules.

grated architecture but not necessarily a modular one.

The characteristics of modular and integrated architectures are compared in Table 1 based on Ulrich and Eppinger (1995). Products are rarely strictly modular or integrated; rather, they involve some degree of modularity.

1. Definition of Modular Products

Modular products refer to products, assemblies and components that fulfill various functions through the combination of distinct building blocks (modules) (Pahl and Beitz, 1988: p. 342). *Modular components* refer to components whose functional, spatial, and other interface characteristics fall within the range of variations allowed by the specified standardized interfaces of a modular product. The mixing and matching of modular components in a modular product design can generate a potentially large number of different products in a modular product model consisting of distinct combinations of components that give each model distinctive functionalities, features, and/or performance levels (Langlois and Robertson, 1992; Sanderson and Uzumeri, 1990; Ward *et al*., 1995). Thus, modular product design is an important form of *strategic flexibility* (Sanchez, 1993), i.e., flexible product designs that allow a company to respond to changing markets and technologies by rapidly and inexpensively creating product variants derived from different combinations of existing or new modular components.

The term *modularity* in products is used to describe the use of common units to create product variants. It arises from the division of a product (part) into independent components, thus allowing one to standardize components and to create a variety of products. Modularity aims to identify of independent, standardized, or interchangeable units to satisfy a variety of functions. With a wide range of overall functions, the partitioning of a product into *function-oriented* modules is of importance while with a small number of overall function variants, a *production-oriented* solution is the paramount consideration (Pahl and Beitz, 1988).

Function modules help to implement technical functions independently or in combination with other functions. Production modules are designed independently of their functions and are based on production considerations alone. Function modules are classified as *basic*, *auxiliary*, *adaptive*, and *non-modules* (Pahl and Beitz, 1988: pp. 343-344).

(1) A *basic module* is a module implementing *basic functions*. The basic functions are not variable in principle and are fundamental to a product or system.

(2) An *auxiliary module* corresponds to *auxiliary functions* that are used in conjunction with the basic modules to create various products.

(3) An *adaptive module* is a module in which *adaptive functions* are implemented. Adaptive functions adapt a part or a system to other products or systems. Adaptive modules handle unpredictable constraints.

(4) A *non-module* implements *customer-specific functions* that do occur even in the most careful design development. Non-modules have to be designed individually for specific tasks to satisfy the customer needs.

Modularity is viewed by Ulrich and Tung (1991) as depending on two characteristics of a design:

(1) similarity between the physical and functional architecture of the design, and

(2) minimization of incidental interactions between physical components.

Based on the interactions within a product, three categories of modularity have been defined (Ulrich and Tung, 1991):

(1) *Component-swapping modularity* occurs when two or more different *basic components* are paired with a module, thus creating different product variants belonging to the same product family.

(2) *Component-sharing modularity* is complementary to component-swapping modularity. Various modules sharing the same *basic component* create different product variants belonging to different product families.

(3) *Bus modularity* occurs when a module can be matched with any number of *basic components*. Bus modularity allows for variation in the number and location of basic components in a product while component-swapping and component-sharing modularity allows only for the types of basic components to vary.

Note that in the above three types of modularity, replacing a basic component with a module interacting with other modules results in different types of module-swapping, module-sharing, or global bus modularity. A customized product may be made up of numerous modules, e.g., the creation of a PC including a terminal, a motherboard, a keyboard modules, etc. As a consequence, a customized product may consist of a base module and several customized auxiliary modules, adaptive modules, or basic components. With this approach, customized products can be produced quickly with lower manufactory costs. The strategy of "modular products design" aims to reduce the design/manufacturing difficulties involved in making customized products. Examples of modularity are presented as follows:

A. Example 1

An example of product variants generated through

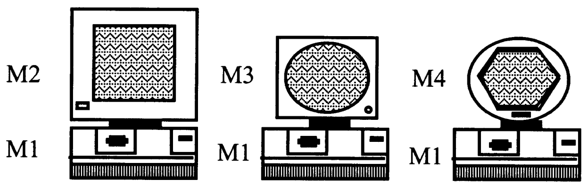

Fig. 2. Example of module-swapping modularity.

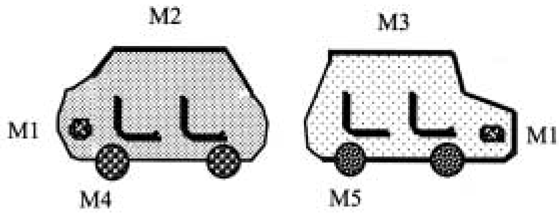

Fig. 3. Example of module-sharing modularity.

module-swapping is provided in Fig. 2. The various computers in Fig. 2 are assembled on a frame board (M1). Different product variants are configured by installing monitors (M2, M3, or M4).

In the automotive industry, by using different audio cassette decks, windshield glass, and wheel types with the same base body of the car, different models of cars are generated. In the computer industry, component (module)-swapping modularity manifests itself through matching of different hard disk types, monitor types, and keyboards with the same motherboard.

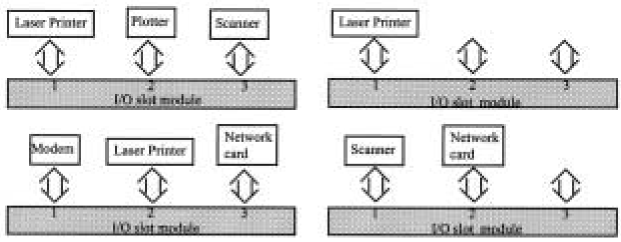

Fig. 4. Example of global bus modularity.

B. Example 2

An example of product variants generated through module-sharing is provided in Fig. 3. The different types of vehicle bodies and tires (M2 and M3, and M4 amd M5) shown in Fig. 3 sharing the same engine (M1) make up different types of cars.

Component (module)-sharing modularity in the automotive manufacturing leads to use of the same brake shoes, alternators, or spark plugs in different product families. In consumer electronics, component (module)-sharing arises when a common power cord or a common tape transport mechanism is used in different product families.

C. Example 3

An example of product variants generated through global bus modularity is provided in Fig. 4. Different types of auxiliary computer equipment, e.g., a laser printer, plotter, scanner, and network card as shown in Fig. 4, are plugged into the same type of I/O slot module, forming computers with different types of functionality.

Other examples of global bus modularity are a computer, circuit breaker, gantry robot, and storage/retrieval system, which use auxiliary components of different types to handle a variety of objects.

2. Industrial Examples of Modular Products

Product variety refers to the range of product models a company can produce within a particular time period to meet the market demand. Products built around modular product architectures can be varied without significant changes in the manufacturing system.

Product variations based on mixing and matching of modular components are now appearing in markets as diverse as aircraft, automobiles, consumer electronics, household appliances, personal computers, software, test instruments, and power tools (Morris and Ferguson, 1993; Sanderson and Uzumeri, 1990; Sanchez, 1991; Sanchez and Sudharshan, 1993).

For example, Swatch produces hundreds of different low cost watch models by assembling the models from different combinations of standard modules (Pine, 1992). A large number of different hands, faces, and wristbands can be combined with a relatively small selection of movements and cases to create seemingly endless combinations.

In the design of the Nippondenso panel meter shown

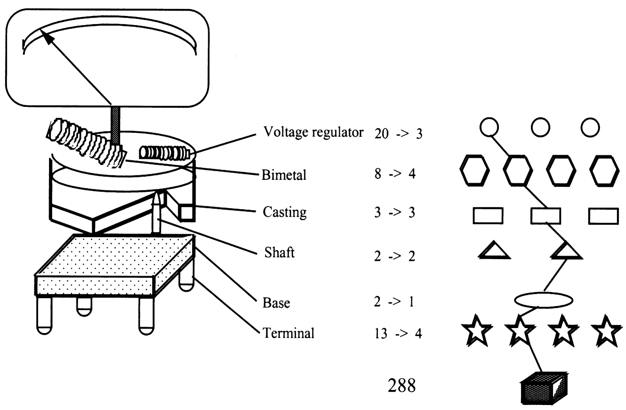

Fig. 5. Nippondenso panel meter.

in Fig. 5, the concept of selectional modularity was applied (Aoki, 1980). The old panel meter design was redesigned to establish six standard modules. The combination of the six modules produces 288 different models, of which about 40 are currently being produced.

3. Benefits and Costs of Modularity

The benefits and costs of product modularity were discussed by Ulrich and Tung (1991). One of the most common motivators for promoting modularity is the need to allow a large variety of products to be constructed from a much smaller set of different modules and components. The result is that any combination of modules and components, as well as the assembly equipment, can be standardized.

Potential benefits of modularity include (Nevins and Whitney, 1989: pp. 56-58; Pahl and Beitz, 1988: pp. 354-355; Corbett *et al.*, 1991):

(1) Economies of scale. Since each module will usually be produced in relatively large quantities, natural economies of scale arise.

(2) Increased feasibility of product/component change. Since each module interface is strictly specified, changes can be made to a module independently of other modules, provided the interfaces remain within specifications.

(3) Increased product variety. The use of modules means that a great product variety can be achieved using different combinations of modules.

(4) Reduced order lead-time. Since modules are manufactured in relatively large volume, the logistics of production can be organized so as to reduce manufacturing lead time. Hence, the order lead time can be reduced.

(5) Decoupling tasks. Since the interfaces and modules have been standardized, their interfaces enable design tasks and production tasks to be decoupled. This decoupling can result in reduced task complexity and in the ability to complete tasks in parallel.

(6) The ease of product upgrade, maintenance, repair, and disposal. Since a product is decomposed into modules, only certain modules need to be replaced when repair is done. For the same reason, upgrades, maintenance, and disposal are also made simpler.

However, potential costs of modularity include:

(1) Redundant physical architecture (due to decreased function sharing).

(2) Excessive capability due to standardization (designing for the most rigorous application).

(3) The potential for static product architectures and excessive product similarity.

III. The Concept of Modular Design

Product designs differ in the degree to which they have been decomposed into *loosely-coupled* (nearly independent) versus *tightly-coupled* (highly independent) components. *Modular design*, as a special form of product design, aims to identify components with a high degree of interaction (Sanchez, 1993).

Design is often defined as the creation of a synthesized solution in the from of products, processes or systems that satisfy perceived needs through mapping between functional requirements (*FR*s) in the functional domain and the design parameters (*DP*s) of the physical domain through the proper selection of *DP*s that satisfy *FR*s (Suh, 1990), i.e., $[FR] = [A] \cdot [DP]$, where $[A]$ is the design matrix. A functional element corresponds to a subsystem (mechanism), and interconnections correspond to function flows in function-oriented modularity. Based on these functions, six types of functional similarity are considered in the identification of modular components: geometric, temporal, force, electrical, thermal, and photometric.

The design of modular products at the conceptual level involves determiniing a design matrix $[A]$ such that the functional requirement space is mapped into the modular functional space. Then, the modular functional space is mapped into the module space based on consideration of module performance, e.g., size, speed, and weight. The mapping among these three different spaces is illustrated in Fig. 6.

The elements of modular functional space are classified as follows (based on Pahl and Beitz (1988)):

BF: basic functions existent in most products, e.g., the power supply in a computer;

AF: auxiliary functions characteristic of variant products resulting from the various types of modularity, e.g., the protection/esthetic function of a lamp cover;

AdF: adaptive functions which are adaptive to differ-

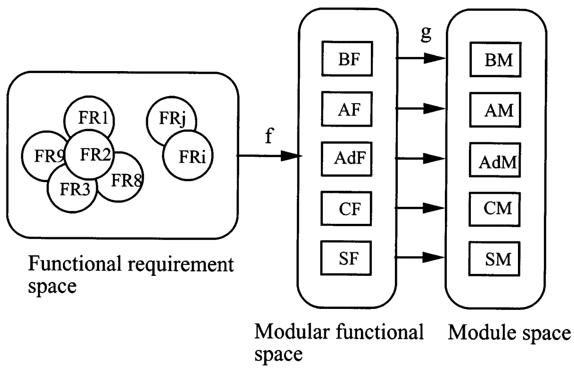

Fig. 6. Mapping in three design spaces.

ent modules/basic components, e.g., the converting function of a computer inference card that standardizes I/O signals;

SF: special functions that may or may not exist, e.g., the eye protection function in a computer product;

CF: customer-specified functions, e.g., the feedback function of vision detection of a missile as specified by the Department of Defense.

The elements of module component space are classified as basic moduless, auxiliary modules, adaptive modules, special module, and custom-specified (non-module) elements as presented in Section II.

Pahl and Beitz (1988) summarized the development of modular products as follows:

Step 1. Clarify the task: Generate specifications. A module normally fulfills several main functions.

Step 2. Establish a functional structure: Subdivide the main functions into a minimum number of similar and recurring subfunctions (BF, AF, AdF, SF, and CF) based on two constraints:

(i) The functional structures of the product variants considered for modularity must be logically and physically compatible.

(ii) The subfunctions determined must be interchangeable.

Step 3. Determine the methodology to be used to implement the subfunctions. Determine solution principles for implementation of the variant subfunctions. Precondition: Look for principles that provide variants without changing working principles and the basic design.

Step 4. Explore the feasibility between interfaces of modules and basic components (geometric, kinematics, and non-motion machine primitives).

Step 5. Review the constraints.

A concept similar to modular design is the "core product" concept. Shirley (1990) examined the problem of redesigning a large product set so as to improve product performance and reduce manufacturing costs. The design features of this "core" product (a prototype) are used to redesign the remaining members of the family. In this way, the design time is reduced. Closely related to the core product concept is the idea of the modular design process. Process modularity makes it possible to handle some aspects of a design independently of other activities. The use of the core product concept and modular design process allows companies to quickly adapt to changes in product and process technologies, and the consumer needs change. By reducing the time and the amount of resources consumed in responding to these changes, system flexibility is enhanced. Moreover, changes can be implemented in a systematic and incremental manner.

1. Difference between Conventional and Modular Designs

The conventional model of the product development process is based on the sequential staging of design tasks (Clark and Weelwright, 1993; Takeuchi and Nonaka, 1986) (Fig. 7).

In the conventional model, after defining the product concept, design activities are typically sequenced so that technology and component development activities with the greatest uncertainty are resolved first. As new technical knowledge is developed and technological uncertainties about components are resolved, design decisions are made, thus allowing the next stage of design activities to be implemented. This process is repeated at each stage of the product development process until all the components and their interfaces are fully specified. Although the product development process may begin with a general idea for the arrangement of components in the design, the actual product architecture, i.e., the full specification of all component interfaces, is determined at the end of the design process. In essence, the product architecture is the output of the sequential design process.

As new component technology and designs leading to component interface specifications are developed sequentially, the need for changes in component designs at an early stage of development may not be discovered until later stages of the design and development of dependent components, are reached, as suggested by the information feedback shown in Fig. 7. If unexpected technical difficulties encountered in developing "downstream" components indicate a need to change "upstream" component designs, intervening component development processes may also have to be replaced in order to accommodate changes made in the upstream component designs, especially those affecting major components. The inherent time delays in this feedback system and the potential high costs involved in recursively redesigning dependent components when

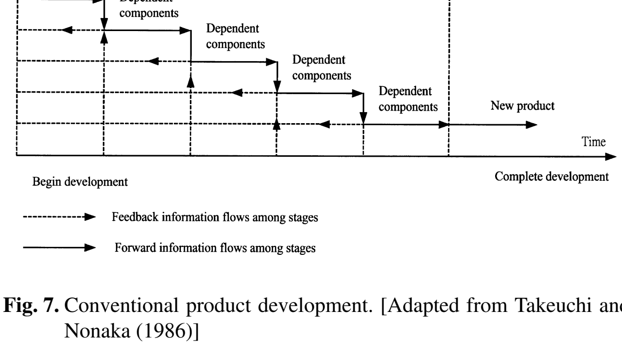

Fig. 7. Conventional product development. [Adapted from Takeuchi and Nonaka (1986)]

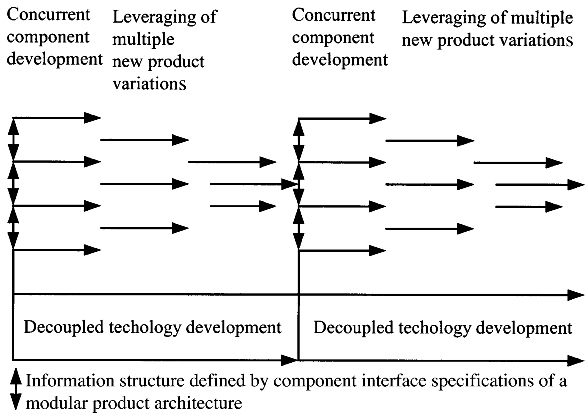

Fig. 8. Development of modular product. [Adapted from Takeuchi and Nonaka (1986)]

changes must be made in the upstream components may reduce the ability of a company to efficiently create and apply technical knowledge about components.

A *modular product design* is one in which input and output relationships between components (i.e., component interfaces) in a product have been fully specified and "standardized". Sanchez (1996) defined *Standardized Component Interfaces* as: functional, spatial, and other relationships between components within a product design that, once specified, are not permitted to change during the product development process and perhaps beyond.

Modular product design implies that there is a new model for managing information flow and knowledge during the product development process (Fig. 8). In contrast to the evolving information structures characteristic of the sequential product development process, a modular product design creates a complete information structure – fully specified component interfaces of a modular product architecture – that defines the desired outputs of development tasks before beginning processes for development and detailed design of components.

Table 2 summarizes the key differences between conventional and modular approaches to defining, designing, and developing new products (Sanchez, 1996).

2. Modular Design in Industry

A. Aircraft Design

In developing the Boeing 777 aircraft, the modular product architecture specified at the beginning of the development process created a positive environment for efficient "localized learning" in developing specific components (Woolsey, 1994). A localized learning environment is possible when development of components can be carried out through autonomous processes. Modular product architectures also provide a framework that supports expanded involvement of lead users in product development (Von Hipple, 1988).

B. Furniture Design

Romero-Subiron and Rosado (1995) described the design of a low cost hierarchical shop floor control system for the modular furniture industry. The system allows for the addition of line resource management modules, such as tools, fixtures, personnel, etc. Those modules that have not been actually developed can be critical to other industries, e.g., the mechanical product industry.

C. Circuit Design

Van den Bout *et al.* (1992) viewed an electrical circuit system as a combination of modules. A module here is a hierarchically nested collection of components and their interconnecting nets. At its lowest level, a module may contain only a single component, e.g., a logic gate. A similar concept is the configurable system concept discussed by Koren and Koren (1991). A dynamically reconfigurable system can change in time without the need to halt the system. For such a system with many sensors or actuators, only a subset of sensors is used at one time. Alternatively, the same hardware is used in different configurations. Stewart and Khosla (1991) proposed the maximum-urgency-first algorithm, which can be used to predictably schedule dynamically changing systems. Mod-

Table 2. Product Definition, Design, and Development Activities in Conventional and Modular Product Design

	Definition	Design	Development
Conventional product design	Attributes of "optimal" product are determined by marketing research.	Desired functionality is decomposed into components, but component interfaces are not specified in details.	Component development and product design co-evolve in a reiterative process. Product architecture is defined by the final product design, i.e., as the output of the development process.
Modular product design	Product is conceived as a platform for leveraging product variations and improved models.	Modular product architecture fully specifies component interfaces and constrains subsequent component development.	Component development processes are concurrent, autonomous, and distributed. Product architecture defined at design stage does not change during development.

ularity supports dynamic system reconfigurability. In developing modular systems, it may be desirable to specify timing constraints on a per-module instead of a per-task basis. For example, a module may consist of two dependent tasks such that the combined worst-case CPU utilization is less than the total utilization of the two tasks.

Kuo and Wang (1992) presented an integrated computer-aided design environment, the VHDL-based Array Reconfiguration (VAR) system, for design, reconfiguration, simulation, and evaluation. The reconfigurable array architecture was used in a modular way to support the simulation and evaluation of different design alternatives by combining fault patterns, fault diagnosis algorithms, reconfiguration algorithms, and reconfigurable architectures.

Sztipanovits *et al*. (1993) discussed structurally adaptive and dynamically reconfigurable systems as important ingredients in the design and development of robust large-scale signal processing systems operating in complex, nonstationary environments.

Fault isolation in an integrated diagnostic environment is another concept similar to the modularity concept. Simpson and Sheppard (1993) discussed a technique which uses a test-to-test matrix to isolate faults under multiple criteria, such as test time, skill level, failure frequency, and information value.

Kelem and Seidel (1992) described a software tool (Z-BLOX) for mapping architecture-independent designs to field-programmable gate arrays. The tool automatically propagates partial data type specifications and performs architecture-specific design optimization and context-dependent module synthesis.

D. Software

The processes of software design involve planning, preliminary design, detailed design, implementation, and testing (Sanchez, 1993). A module is usually formed in the planning and preliminary design process, and is implemented with objects in object-oriented programming, e.g., C++ and Java, in the implementation process (Coplien and Schmidt, 1995). In software design, a broader definition of a module was given by Yourdon and Constantine, namely that, “A module is a lexically contiguous sequence of program statements, bounded by boundary elements, having an aggregate identifier” (Yourdon and Constantine, 1979). Examples of boundary elements are packages (“begin... end pairs”) in a block-structured language like Pascal or Ada (Walker, 1993; Schach, 1993) or objects (“{...}” pairs) in object-oriented languages like C++ and Java (Fisher, 1991).

Such an object-oriented approach takes advantage of some modularity characteristics. However, there is now a widespread belief that software engineering must go beyond object oriented methods to a new technology based upon “architecture” (Luckham and Vera, 1995). Technologies such as the Common Object Request Broker Architecture (CORBA), for example, allow distributed systems of interacting modules to be wired together. However, an architecture plan for the system is needed to both guide “wiring-up” and to prototype the behavior of the system before any “effort” is put into building the modules (i.e., the system’s components).

Modularity in software has a distinct implementation flavor, being concerned with the allocation of objects and functionality to compilation units (Walker, 1993), e.g., the MODULE language. In all but the simplest applications, there is coarse granularity that is superordinated to the highest level objects; it is related to the decomposition of the application into sub-domains (somewhat distantly related to Coad and Yourdon’s subjects (Coad and Yourdon, 1990)). Below the level of sub-domains, modularity rests on the same foundation as, in part, does abstraction; the encapsulation of code and data within the object, within hierarchical composition and class structures. In conventional programming terms, the module is an integrated component used to process object identification and encapsulation when the target implementation language has suitable mechanisms for refecting the design structure in the implementation (Walker, 1993). However, there is a problem here, since, in many object-oriented programming languages, classes are grouped within the class hierarchy (a-kind-of inheritance hierarchy) and there is no explicit grouping that corresponds to the composition/aggregation hierarchy. Nevertheless, it is perhaps the latter case that most closely corresponds to the designer’s intuitive understanding of the problem domain. Modularity should reflect the cohesion of the application domain in some circumstances whereas in others, it might be required to reflect construction/maintenance issues, e.g., keep all hardware-dependent functions localized in a few classes that are grouped together. A methodology should address these diverse aspects of modularity.

3. Modular Manufacturing

Tsukune *et al*. (1993) discussed the characteristics of future manufacturing systems and proposed a concept of modular manufacturing which could be used to integrate intelligent machines. In large-scale manufacturing systems, modularization is indispensable for clarifying the logical structure and assuring a high degree of ease of construction. The parts, products and manufacturing equipment as well as the design and operating activities themselves are all described in units called modules. A manufacturing system is constructed and operated by combining these building blocks. Hardware and software modules are combined to meet specific requirements.

Stoll (1986) noted that modular construction enables 'standardized diversity' by using different combinations of standard components. Modular design resists obsolescence, shortens redesign, enables new designs to be realized using existing modules, reduces costs and eases maintenance. Moreover, where modular construction methods have become widely established, such as in electronics using standard components, e.g., ICs, resistors, and capacitors, the design process is generally assisted by more sophisticated design and verification tools. With regard to manufacturing equipment, the modular concept has been used for many years. For example, many machine tool manufacturers produce customized machine tools largely by configuring their existing machine subsystems. Automation equipment suppliers, such as Festo, SMC and Parker, supply proprietary modular hardware, e.g., actuators units and grippers, for building modular work handling systems. Moreover, although traditionally associated with hard automation, the selectional modular approach is gaining recognition as an alternative means of achieving the promised flexibility of anthropomorphic robots, albeit by means of configuration (Weston *et al.*, 1989).

Modular concepts are also used in the construction of manufacturing systems. Noted benefits of this approach include the following:

(1) They provide greater scope in the way production is organized and the opportunity to readily reconfigure production to meet both short and long term objectives (Merchant, 1985).
(2) They simplify the integration of processes, machine systems, tooling, people, organizational structures, information flows, control and computer systems necessary to perform a given task (Davis, 1991).
(3) They help eliminate islands of automation and further the reuse of machinery (Tsukune *et al.*, 1993).

To build, test, and calibrate a conventional general-purpose oscilloscope at Philips typically takes eight weeks; that also includes stuffing the printed-circuit boards. Its new manufacturing techniques reduced this process to only 5.5 days. Assembling an oscilloscope takes only 20 minutes. Due to the one-piece chassis, only relatively few parts are needed to mount modules, and PC boards as well as to perform computerized test procedures. In contrast, it takes on average 10 hours to assemble other models in the same company. The Philips chassis comes equipped with most of the click fit mountings, snap-in fixtures, stops, posts, and guides needed to mount or insert into the instrument eight basic functional modules: an activator, a vertical amplifier, X- and Y- amplifiers, a time-based unit, a cathode-ray-tube control, a power supply, a front-panel unit, and a liquid-crystal display unit. Only a few nonintegrated fixtures are needed to retain cables and to hold certain other parts in place. Another speed-enhancing measure – the use of an automated module test system on the manufacturing line – ensures high product quality. Testing at module levels – interim testing, as it is called – ensures that no faulty modules are ever built into an instrument. Interim testing also cuts the final test and calibration time.

Tsukune *et al.* (1993) noted that there are significant problems currently limiting progress in modular manufacturing, namely:

(1) the large number of manufacturing machine elements currently in use makes modular manufacturing system design and control difficult;
(2) the design, manufacturing and control processes are based on completely different models, which results in complex transformations between the 'idealized world' in which design tools operate and the 'real-world' in which manufacturing occurs.

The problems stem principally from the fact there are no standards for modular equipment. Moreover, there is no agreement on what the building elements should be. For example, a machine module currently encompasses everything from a complete machine tool, such as a robot with an integrated controller, to a machine building element, such as an actuator unit, motor or transmission system. Indeed, such diversity of manufacturing machinery exists principally as a result of the wide diversity of production requirements.

The modular manufacturing systems (MMS) in Japan are aimed specifically at low to medium level technology consumer products, as typified by goods such as children's toys and kitchen appliances (Tsukune *et al.*, 1993). The rationale for MMS as a means of enabling concurrent product and manufacturing system design has been put forward and the long term implications and work required to establish the concept have been discussed (Stoll, 1986). The MMS concept has been proposed as a way of overcoming limitations resulting from a lack of modular machine standards (Rogers, 1990). Moreover, MMSs seek to provide a radical new manufacturing business framework suitable for the 'agile' manufacturing era. The module standards are based on a unified 'reduced' set of 'primitive' production elements. The module categories are comprised of four classes, namely process machine primitives, motion units, modular fixturing and configurable control systems. Appropriate selection of modules from these categories should make it possible for a diverse range of efficient, automated and integrated production systems to be built.

The modules in a product architecture are analogous to flexible cells or holons in manufacturing systems. A cell in the Group Technology (GT) is dedicated to a certain group of workpieces. The cellular system is likely to give the best match of machining capacity to process time for various workpieces. It also lends itself to the gradual introduction of Flexible Manufacturing Systems for different types of workpieces (Hartly, 1984). Holons, conceptually, are similar to biological cells with autonomous, distributed, and cooperative characteristics (van Leeuwen and Norrie, 1997). Holonic systems with the characteristics of intelligence and adaptability aim to achieve integration and optimization of manufacturing systems in order to produce products at lower cost and higher quality and efficiency. A product module may be produced in a cell (holon) or a group of cells (holons). A cell or holon may manufacture a variety of modules. Such systems possess flexibility and agility. Both flexible cells and holons are *production-oriented* resolution (see Section I).

Information on the World Wide Web (WWW) on modular products and companies which employ in modular design practices is listed in the Appendix.

IV. Research Issues in the Development of Modular Products

Modular product development is becoming a focus of attention and is frequently stated as a goal of good design practice. However, it has not received sufficient attention in the literature (Pahl and Beitz, 1988; Shirley, 1990; Ulrich and Tung, 1991). It has not been explored in industry to the same degree as, for example, design for manufacturing. Pahl and Beitz (1988) and Ulrich and Tung (1991) have presented numerous fundamentals.

Besides the works presented in Sections II and III, the significant related works include the following: Kusiak and Huang (1996) developed a methodology for determining modular products while considering cost and performance. To interpret various types of modularity such as component-swapping, component-sharing, and bus modularity, a graphical representation of the product modularity was presented while the module components of a product set were determined using a heuristic approach. With the module components known, a rule-based fuzzy representation of the module development problem was presented while the tradeoff between performance and module cost was analyzed using a fuzzy neural network approach. The approach was illustrated with an example of a multichip module.

Kusiak and Huang (1997) applied the concept of modularity to the development of modular products and product testing in modular tests. The relationship between the design of modular products and testability, and the testing of products in modular tests was explored. Methodologies for the design of modular products for testability and the design of testing modules were developed. An integrated system for the design of modular products and test processes was presented.

Huang and Kusiak (1998) developed the models and solution approaches to the modularity problem for mechanical, electrical and mixed process products (e.g., electro-mechanical products).

Huang and Kusiak (1999) presented a module-based design approach to mechatronic products with consideration of performance criteria, e.g., testability of electronic subsystems. In this research, a generalized LC algorithm was developed which is useful in various ways to determine testable values, points, and paths.

Table 3 summarizes related previous works and compares their models, methodologies, and the challenges faced. Corresponding to these challenges, potential research issues are presented in this section. Future possible research is classified based on the following five issues: the representation and formulation of modularity, module size, the knowledge management and collaborative design of the modular products, virtual reality in the design of modular products, and the impact of modularity on manufacturing systems.

1. Representation and Formulation of Modularity

Modules ideally should be formed early in the design process, e.g., at the conceptual design phase. However, the information needed to identify the modules may not be available. As a result, modules generated early in the design process may not satisfy the constraints that become apparent later in the design process. In this situation, a proper representation of modularity is crucial.

Furthermore, the goals of modularity in product design may vary; e.g., the goal of designing Multichip Modules (MCMs) in electronics is to decrease the spacing between integrated circuits (ICs) (Kota and Ward, 1990) rather than increase the variety of product characteristics as in agile manufacturing. However, MCM products are more expensive than an equivalent collection of single chip components and PCBs. Products are often made to order. To increase the potential for using MCMs at reduced design and manufacturing cost, a methodology needs to be developed to identify modules in way of increasing the product variety.

A module is usually generated as a result of tradeoffs among the cost and performance attributes, such as size, weight, or speed. The analysis of cost and performance tradeoffs is a complex task; specifically, the performance attributes are only described in fuzzy terms early in the design process, e.g., small size, light weight, and high speed. From the performance point of view, the wide variety of design options available today to the design engineer preclude an exhaustive analysis of all viable alternatives. From the cost perspective, the analysis usually is even more cursory because of the complexity and uncertainty of cost before actual production. For example, up to 80% of the cost of a product could be determined in

Table 3. Summary of Previous Works

Reference	Representation model of modules/ components	Development methodology	Focus on	Challenges
Sanchez (1993), and Pahl and Beitz (1988)	Function Tree	Top-down analysis	Functional decomposition	No formal approaches to determining modules.
Morris and Ferguson (1993), Sztipanovits *et al*. (1993), Pine (1992), Kelem and Seidel (1992), and Sanchez (1991)	None	Rules of thumb	Individual products range from aircraft to automobiles, consumer electronics, household appliances, personal computers, software, test instruments, and power tools	No formal approaches to determining modules, representing modularity, optimizing modular designs.
Shirley (1990) Walker (1993), Stephen (1993)	Core product Package or procedures	Rules of thumb Jackson system development (KSD)	Process modularity Software design	No formal approaches to determining modules. How big should a module be? How complex is this module? How can we minimize interactions between modules? Reuse of modules.
Fisher (1991), Coplien and Schmidt (1995)	Object	Object-oriented approach, CASE	Software design	How big should a module be? How complex is this module? How can we minimize interactions between modules? Reuse of modules.
Tsukune *et al*. (1993), Tsukune (1993)	Manufacturing modules including parts, products and manufacturing equipment all described in units	Maximize degree of ease of construction	Modular manufacturing	The relationship between modular manufacturing and modular design needs to be explored. The large number of manufacturing machine elements makes modular manufacturing system design and control difficult. The design, manufacturing and control processes are based on completely different models; this results in complex transformations between the 'idealized world' in which design tools operate and the 'real-world' in which manufacturing occurs.
Suh (1990)	Design matrix [*A*]	$[FR] = [A] \bullet [DP]$	Mapping between functional requirements (FRs) in the functional domain and the design parameters (DPs) of the physical domain	Size of module. Fuzzy requirement of customer.
Stoll (1986)	Standard components	None	Reduces costs and eases maintenance	No formal approach.
Kusiak and Huang (1996)	Graphical representation of the product modularityand a rule-based fuzzy representation	A heuristic approach	Determining modular products while considering cost and performance	The generic fuzzy rules need to be defined by experts. The input performance attributes need to be selected based on the characteristics of different types of products. The determination of the threshold index to bound the size of modules.
Kusiak and Huang (1997)	Modularity matrix and interaction matrix	Decomposition approach based on expert system of testability	Relationship between the design of modular products and testability	The integration of the decomposition approach with the current design and manufacturing systems. An expert system supporting the identification of testability points needs to be developed. Trade-off analysis of the design guidelines for testability should be developed.
Huang and Kusiak (1998)	Matrix representation	Decomposition approach	The modularity problem for mechanical, electrical and mixed process products	Approaches to optimizing modular designs, and the assessment of the impact of modularity on the design process, manufacturing, and management need to be developed.
Haung and Kusiak (1999)	Logic network	A generalized label-correct (GLC) algorithm	Mechatronic products	The specified design model and synthesis to fit particular products/design systems need to be developed. An expert system supporting the guidelines used to identify testability points needs to be developed. Trade-off analysis of the design guidelines for testability should be developed.

the design phase (O'Gruin, 1990). Hence, the determination of good alternatives which satisfy the performance attributes with some fuzzy constraints at a reasonable cost (called the module development problem) is crucial in building modules.

2. The Module Size

The size of a module is often compromised due to several factors, e.g., the panel size, performance requirements, cost, or testability.

Khan and Madisetti (1995) presented a systematic approach to partitioning multichip-based systems for low power, considering the area of panel, yield, reuse of cell libraries, and routing constraints.

To date, growth in size and complexity has made testing of chips and multichip modules difficult and expensive. Indeed, the cost of testing electronic products is growing more rapidly than is that of other components (Trischler and Johansson, 1994). Furthermore, minimizing the product development time requires the integration of design and testing at early stages of product development. Hence, the testability design has become another crucial factor in partitioning a circuit and determining the boundary of a module.

In conventional design, a circuit is partitioned into reasonably small functional blocks (clusters). In the modular design approach, a block (cluster) corresponds to a module. The partitioning problem is concerned with breaking a circuit into modules such that the partition makes the circuit system easier to understand, easier to control by including reasonably direct paths from the test resources (either automatic test equipment or built-in-test circuitry) to critical internal nodes required for initialization of the circuitry being tested, and partitioning that circuitry, and easier to control for fault activation. The procedure for forming modules with testability considered may involve, besides partitioning, adding control and visible points. Testability has become an important constraint in the module formation problem.

A. Example 4: Consider a Six Level Logic Chain in Fig. 9.

The result of partitioning and adding control points (CPs) and visible points (VPs) to the six level logic chain by adding gates and control points to break long chains and cut programming time for testability is illustrated in Fig. 10 (Turino, 1990: p. 9).

Added CPs and VPs break long chains and cut programming time. Three modules are determined: {U1, U3, U4, CP4, VP1}, {U9, U10, U13, CP1}, {U11, U12, U14, U15, U2, U5, CP2, CP3, CP5, VP2, VP3}.

Using alternative control points and visible points may result in modules of different sizes as shown in Fig. 11, where the placement of CP3 has changed. Three modules are determined: {U1, U3, U4, CP4, VP1}, {U9, U10, U13, CP1}, {U11, U12, U14, U15, U2, U5, U6, U8, U18, VP2, VP3, VP4}, where the third module is larger than the one in the previous partition. The different placement of CPs and VPs may result in modules of different sizes.

An airplane can be made lighter, however, probably at the expense of increased manufacturing cost. One of the most difficult aspects of product development is recognizing, understanding, and managing such trade-offs in a way that maximizes the success of the product. Trade-

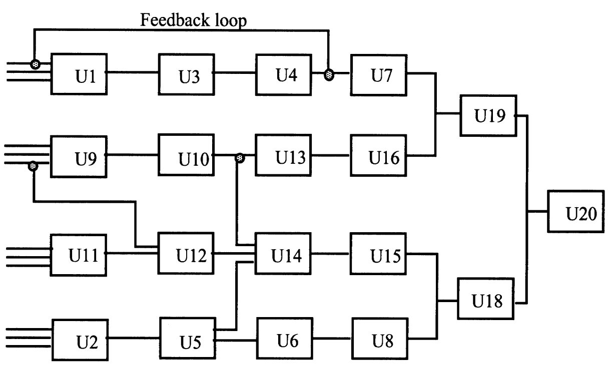

Fig. 9. Six level logic chain.

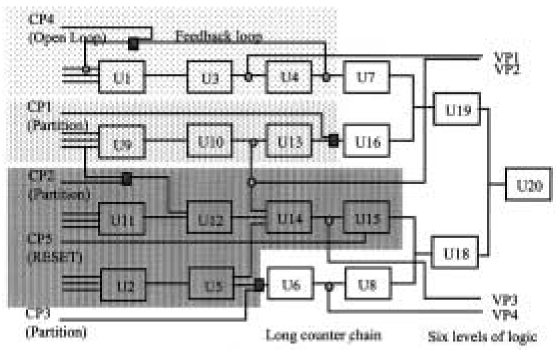

Fig. 10. The result with CPs and VPs added.

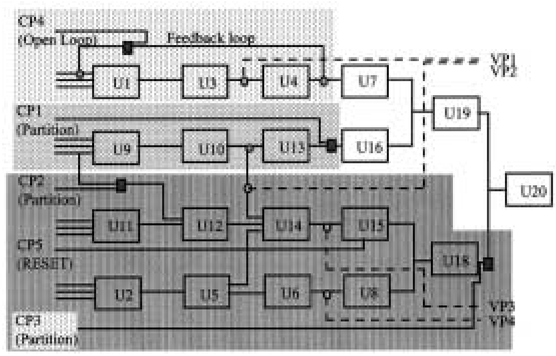

Fig. 11. The alternative result with more CPs and VPs added.

off analysis between the size and performance, e.g., testability, of modules needs to be studied.

3. Knowledge Management and Remote Collaboration in Design of Modular Products

Design of modular products implies a new model for managing information flow. To fully specify and standardize component interfaces in the design of modular products, knowledge about relevant components and their interactions is required. In essence, a modular design must be based on knowledge about relevant components and their interactions in which developers can justifiably have a high level of confidence. A modular design process tries therefore, to leverage existing knowledge in which participating companies have a high level of confidence by developing and designing components that conform to an adopted, fully specified product architecture. As a result, processes for applying existing knowledge in product design must be intentionally separated from and only loosely coupled to the process of creating new technical knowledge that could affect component interface specifications during a given design project. The issues involved in knowledge management include the database representation of modules and the search approaches applied to the database, especially in a distributed system.

To speed up the product design process, it is not necessary to design each individual module but to rather use modules that have been created. Remote collaborative design through the Internet appears to be feasible due to the rapid development of WWW technology. Distributed collaborative design of modular products provides a more effective way to respond to changing market requirements. The Internet is the world wide network based on the Transmission Control Protocol/Internet Protocol (TCP/IP) standards. The WWW is a very large collection of clients and servers that support the Hyper Text Transfer Protocol (HTTP). This is an open standard and is implemented on a wide variety of platforms. The popularity of the WWW arises from its being accessible on a wide range of platforms, from the ease of moving information from one platform to another, and from its graphical user interface (usually called a browser). The client uses a browser to form a request, send it to a server, and receive the results from the server. A server receives and validates the request, retrieves data, and delivers them to the requesting client. The result is a mechanism used to share data with little regard to distance or to the different computer platforms in use.

Some research has been done to facilitate remote collaboration. For example, FLECSE (Flexible Environment for Collaborative Software Engineering) is a multimedia environment designed to facilitate the communication between two or more geographically dispersed software engineers (Dewan and Riedle, 1993). Unfortunately, there is no work reported, as yet, that examines collaboration with suppliers in general, or using the Internet in particular, or indeed the area of designing with modules. Modular product design would appear to be an attractive proposition, but little work has been done on these research issues (Pahl and Beitz, 1988; Shirley, 1990; Ulrich and Tung, 1991) particularly for the relatively common design environment in which modules come from geographically separate locations and differing computing platforms are used by those involved.

The solution should be able to address the resulting research issue:

> How can designing with modules be carried out to meet customer requirements using modules that come from suppliers that may be geographically separated, and that may operate on differing computer platforms?

4. Virtual Reality in the Design of Modular Products

In agile manufacturing, product presentation is crucial, specifically through collaborative design. The existing technique uses the Virtual Reality Modeling Language (VRML). However, it has two basic limitations (Vacca, 1996; Matsuba and Roehl, 1997: pp. 557-594) which hinder distributed collaborative design:

(1) it is fundamentally static;

(2) it lacks interaction, animation, and behavioral capabilities

There is currently no provision for what is sometime called "object behavior" in VRML. This concept refers to the assignment of properties to objects that would define how they would behave, whether in motion by themselves, or in combination with other objects. A good example of this property for objects of the class "ball" might be "bouncing." Balls made of hard rubber (like a tennis ball) will bounce longer and higher than balls made of foam (like Nerf balls). An important part of animation in a 3-D world depends on the description and modeling of such behavior of objects.

Given that most mechanical, electrical and other systems are already designed using CAD software (and hence, are already available as 3-D models), VRML is expected to be extended and to enable, for example, disassembly and reassembly of virtual parts of motors, equipment, vehicles etc. The limitations of VRML are the issues which need to be studied in distributed collaborative design.

5. The Impact of Modular Products on Manufacturing and Concurrent Engineering

The impact of modular products on the performance of manufacturing systems should be thoroughly analyzed. For example, in the case of assembly design, Boothroyd (1991) suggested the use of the minimum part count rule. Simplification of the product structure can lead to substantial savings in the assembly cost of parts. After examining some specific cases where problems have arisen from the application of the minimum part count rule, Barkan and Hinckley (1993) opposed rigid adherence to the minimum part count rule. In several instances, the large part count facilitated significantly simpler part fabrication as well as simpler assembly operations. They suggested that the implication of design for manufacturing process rules should be examined in a broad context. He and Kusiak (1996) studied the impact of modular products designs on the performance of manufacturing systems. The performance of product designs was measured based on the makespan of corresponding aggregate schedule of the manufacturing system.

Decomposition, standardization, and exchangeability are the attributes of a modular product. Decomposition enhances the controlability and observability of testing, respectively, thus reducing the complexity of the testing process. For example, assume that a test on a circuit with n gates involves n^2 steps, and that circuit C with 10,000 gates and can be partitioned into two circuits, C_1 and C_2, of 5,000 gates each. Then, the test generation for the unpartitioned version of C requires $(10^4)^2 = 10^8$ steps, while the test generation for C_1 and C_2 requires only $2 \times 25 \times 10^6 = 5 \times 10^7$ or half of the testing time. Standardization simplifies test adapter designs and also reduces the number of different test adapters requireds for any given system. Similar to using standard pin configurations, standardization of connector types reduces the number of types of test adapters required and improves manufacturing and logistical conditions. The relationship between modular design and manufacturing needs to be explored.

In most cases, modules are formed in the conceptual design process. To make use of the modularity characteristics, a module should integrate with a manufacturing cell in GT, a honolic cell in agile manufacturing, or an information object in object-oriented design. The way to integrate a product module with a manufacturing cell, a honolic cell, or an information object is to use *Concurrent Engineering*, where, ideally, functions such as design, manufacturing and information functions are integrated, resulting in a corporate structure that can manufacture and design products at lower cost, at improved quality, and in a shorter time. The development of optimal integration is crucial and needs to be studied in the future.

V. Conclusion

While customized products are on track and the manufacturing industry is undergoing a major paradigm shift into a world of agility, the further requirements of wide variety and rapid product development are gradually being imposed. The modular approach promises the benefits of high volume production (arising from producing standard modules) while, at the same time, being able to produce a wide variety of products that are customized for individual customers. Such modular product design has been stated as a goal of good design practice in the concurrent engineering area. However, it has not received sufficient attention in the literature. This paper has presented an overview of modular product development. The significance aspects of this work include:

(1) a review of the concept of modular products and modularity;
(2) an indroduction to the definitions of modular product and modularity;
(3) discussion of the design of modular products, software, and manufacturing systems;
(4) illustration of modularity using industrial examples;
(5) exploration of the differences between modular and traditional product designs;
(6) a survey of the literature.

Through a literature survey, the challenges faced in the development of modular products have been clarified. Based on the unclear, the research issues can be summarized as follows:

(1) the representation of modularity and determination of modules;
(2) trade-off analysis between module size and performance;
(3) knowledge management and remote collaboration;
(4) the use of virtual reality tools in collaborative design of modular products;
(5) the impact of modular product design on manufacturing processes and systems.

The main contribution of this paper have been:

(1) to present the concept of modularity to those who are not familiar with the concept and the importance of modularity;
(2) a review of the literature on modular product design and the formulation of research issues related to the development of modular products for those who are working in this field.

Acknowledgment

This work was partially supported by funding from the National Science Council of the Republic of China (NSC 88-2213-E-260-003).

Appendix
Web Sites of Modular Product Design

The URL locations of information on the WWW related to modular product design:

PEP Modular Computers Business Center
-- http://www.industry.net/pep.modular
Modular Solutions
-- http://www.mod.com/solution.html
Modular Products
-- http://www.mod.com/products.html
Modular Shelters & Glazing Products
-- http://www.netis.com/members/modular/
MODULAR MACHINE INC. – Tool & Die Engineering, based in Michigan, Iosco County
-- http://host.pc.centuryinter.net/modular/
Research Summary
-- http://robby.caltech.edu/~chen/research.html
Northwest Modular
-- http://www.daka.com/nwmod/
Advanced Modular Solution's Web Site
-- http://www.mod.com/
Tandem Products
-- http://www.tandem.com/INFOCTR/HTML/PROD_DES/SUFOSWPD.html
Modular Form
-- http://www.electriciti.com/balloon/modular.html
The Modular Way
-- http://www.mod.com/modway.html
Allied Modular Building System Business Center
-- http://www.industry.net/allied.modular.building
Modular Protection Corporation
-- http://www.indra.com/unicom/company/modularpro.html
Westchester Modular Homes
-- http://www.qgm.com/westchester/
TOSHIBA - The World's Best Selling Portable Computers.
-- http://www.toshiba.com/tais/csd/products/400s.htm
PCD-5ND
-- http://www.sni.de/public/pc/pc_prod/infos/epcd5nd.htm
Robitech's Home Page
-- http://www.robitech.com/
Product Index
-- http://www.crosscomm.com/product.html
MMDS1632
-- http://freeware.aus.sps.mot.com/dev_tools/intr1632.html
Overview of EDGECAM
-- http://mfginfo.com/cadcam/edgecam/edgecam.htm
Matrix-UPS
-- http://www.apcc.com/matrix.htm
Labyrinth Modular Clothing
-- http://wwbc.com/english//profiles/laby/laby.html
Some mathematical ideas from modular arithmetic used in RSA
-- http://rschp2.anu.edu.au:8080/modulus.html
Apex Data Home Page
-- http://www.apexdata.com/
Gateway 2000 Solo
-- http://www.gw2k.com/product/portable/solo/sololist.htm
The Power of Modularity
-- http://www.mod.com/modpower.html
Modular Company Backgrounder
-- http://www.mod.com/modback.html
Modular Wall Panel
-- http://www.invention.com/buster.htm
Classic Modular Systems Incorporated
-- http://www.dataplusnet.com/cms/cmsmain.html
Quality Modular Interior Services
-- http://inet1.inetworld.net/~dcs/Qmod/
Classic Modular Systems
-- http://www.dct.com/cms/
Modular SuperBank Systems
-- http://www.cpgs.com/msbs/
Star Modular Industries Home Page
-- http://www.patiostar.com/

References

Aoki, K. (1980) High speed and flexible automated assembly line – Why has automation successfully advanced in Japan. 4th International Conference on Production Engineering, Japan Society of Precision Engineering, Tokyo, Japan.

Barkan, P. and C. M. Hinckley (1993) The benefits and limitations of structured design methodologies. *Manufacturing Review*, **6**(3), 21-220.

Boothroyd, G. (1991) *Assembly Automation and Production Design for Assembly*. Mercel Dekker, New York, NY, U.S.A.

Clark, K. B. and S. C. Wheelwright (1993) *Managing New Product and Process Development*. Free Press, New York, NY, U.S.A.

Coad, P. and E. Yourdon (1990) *Object-oriented Analysis*. Prentice Hall, Englewood Cliffs, NJ, U.S.A.

Coplien, J. O. and D. C. Schmidt (1995) *Pattern Language of Program Design*. Addison-Wiley, Reading, MA, U.S.A.

Corbett, J., M. Dooner, J. Meleka, and C. Pym (1991) *Design for Manufacturing: Strategies, Principles, and Techniques*. Addison Wesley, New York, NY, U.S.A.

Davidow, W. and M. Malone (1992) *The Virtual Corporation*. Harper Collins, New York, NY, U.S.A.

Davis, R. K. (1991) A system approach to machinery design and implementation. *Proceedings of Eurotech Direct, Conference on Mechanical Engineering: Machine Systems*, **I**, 19-24.

Dewan, P. and J. Riedl (1993) Toward computer-supported concurrent software engineering. *Computer*, **26**(1), 12-16.

Fisher, A. S. (1991) *CASE Using Software Development Tools*. Wiley, New York, NY, U.S.A.

Hartly, J. (1984) *FMS at Work*. Wiley, New York, NY, U.S.A.

He, D. W. and A. Kusiak (1996) Performance analysis of modular products. *International Journal of Production Researches*, **34**(1), 253-272.

Huang, C. C. and A. Kusiak, (1999) Synthesis of modular mechatronic products: a testability perspective. *IEEE Transactions on Mechanics* (accepted).

Huang, C. C. and A. Kusiak (1998) Modularity in design of products. *IEEE Transactions on Systems, Man, and Cybernetics, A*, **28**(1), 66-77.

Kelem, S. H. and J. P. Seidel (1992) Shortening the design cycle for programming logic devices. *IEEE Design & Test of Computers*, **9**(4), 40-50.

Khan, A. K. and V. K. Madisetti (1995) System partitioning of MCMs for low power. *IEEE Design & Test of Computers*, **12**(1), 41-52.

Kidd, P. T. (1994) *Agile Manufacturing: Forging New Frontiers*. Addison Wiley, New York, NY, U.S.A.

Koren, I. and Z. Koren (1991) Discrete and continuous models for the performance of reconfigurable multistage systems. *IEEE Transactions on Computers*, **40**(9), 1024-1033.

Kota, S. and A. C. Ward (1990) Functions, structures, and constraints in conceptual design. *Proceedings of the 2nd International Conference on Design Theory and Methodology*, Chicago, IL, U.S.A.

Kuo, S. Y. and K. Wang (1992) Computer-aided modeling and evaluation of reconfigurable VLSI processor arrays with VHDL. *IEEE Transactions on Computer Aided Design*, **11**(2), 185-197.

Kusiak, A. and C. C. Huang (1996) Development of modular products. *IEEE Transactions on Components, Packaging, and Manufacturing Technology, A*, **19**(4), 523-538.

Kusiak, A. and C. C. Huang (1997) Design of modular digital circuits for testability. *IEEE Transactions on Components, Packaging, and Manufacturing Technology, C*, **20**(1), 48-57.

Langlois, R. N. and P. L. Robertson (1992) Networks and innovation in a modular system: lessons from the microcomputer and stereo component industries. *Research Policy*, **21**(4), 297-313.

Luckham, D. C. and J. Vera (1995) An event-based architecture definition language. *IEEE Transactions on Software Engineering,* **21**(9), 717-734.

Matsuba, S. N. and B. Roehl (1997) *Using VRML*. Que, Indianapolis, IL, U.S.A.

Merchant, E. (1985) The important of flexible manufacturing systems to the realization of full computer integrated manufacturing. In: *Flexible Manufacturing Systems*. H.J. Warnecke Ed. IFS & Spinger-Verlag, Bedford, U.K.

Morris, C. R. and C. F. Ferguson (1993) How architecture wins technology wars. *Harvard Business Review*, **71**(2), 86-96.

Nevins, J. L. and D. E. Whitney (1989) *Concurrent Design of Products & Processes: a Strategy for the Next Generation in Manufacturing*. McGraw-Hill, New York, NY, U.S.A.

O'Gruin, M. C. (1990) Activity-based costing: unlocking our competitive edge. *Manufacturing Systems*, **8**(12), 35-39.

Pahl, G. and W. Beitz (1988) *Engineering Design*. The Design Council, London, U.K.

Pine, B. J., II (1992) *Mass Customization: The New Frontier in Business Competition*. Harvard Business School Press, Boston, MA, U.S.A.

Rogers, G. G. (1990) *Modular Production Systems: a Control Scheme for Actuators*. Ph.D. Dissertation. Loughborough University, Loughborough, U.K.

Romero-Subiron, F. and P. Rosado (1995) The design of a line control system for the modular furniture industry. *International Journal of Production Research*, **33**(7), 1953-1972.

Sanchez, R. (1991) *Strategic Flexibility, Real Options, and Product-based Strategy*. Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge, MA, U.S.A.

Sanchez, R. (1993) Strategic flexibility, firm organization, and managerial work in dynamic markets: a strategic options perspective. *Advanced in Strategic Management*, **9**, 251-291.

Sanchez, R. (1996) Strategic product creation: managing new interactions of technology, markets, and organizations. *European Management Journal*. **14**(2), 121-138.

Sanchez, R. and D. Sudharshan (1993) Real-time market research. *Marketing Intelligent and Planning*, **11**(8), 29-38.

Sanderson, S. W. and V. Uzumeri (1990) *Strategies for New Product Development and Renewal: Design-based Incrementalsim*. Working Paper, Center for Science and Technology Policy, Rensselaer Polytechnic Institute, Troy, New York, NY, U.S.A.

Schach, S. R. (1993) *Software Engineering*. IRWIN, Burr Ridge, IL, U.S.A.

Shirley, G. V. (1990) Models for managing the redesign and manufacture of product sets. *Journal of Manufacturing and Operations Management*, **3**(1), 85-104.

Simpson, W. R. and J. W. Sheppard (1993) Fault isolation in an integrated diagnostic environment. *IEEE Design & Test of Computers*, **8**(1), 52-65.

Stewart, D. B. and P. K. Khosla (1991) Real-time scheduling of dynamically reconfigurable system. IEEE Conference on Systems Engineering, Fairborn, OH, U.S.A.

Stoll, H. S. (1986) Design for manufacture: an overview. *ASME Applied Mechanics Reviews*, **39**(9), 1356-1364.

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

Sztipanovits, J., D. M. Wilkes, and G. Karsai (1993) The multigraph and structural adaptivity. *IEEE Transactions on Signal Processing*, **41**(8), 1695-1716.

Takeuchi, H. and I. Nonaka (1986) The new product development game. *Harvard Business Review*, **62**(1), 34-39.

Trischler, E. and M. Johansson (1994) Ten: a concurrent test engineering environment. *IEEE Design & Test of Computers*, **11**(3), 6-16.

Tsukune, H., M. Tsukamoto, T. Matsushita, F. Tomita, K. Okada, T. Ogasawara, K. Takase, and T. Yuba (1993) Modular manufacturing. *Journal of Intelligent Manufacturing*, **4**(2), 163-181.

Turino, J. (1990) *Design to Test*. Voan Nostrand Reinbold, New York, NY, U.S.A.

Ulrich, K. and K. Tung (1991) Fundamentals of product modularity. In: *Issues in Design/Manufacture Integration 1991*, pp. 73-79. A. Sharon Ed. ASME, New York, NY, U.S.A.

Ulrich, K. T. and S. D. Eppinger (1995) *Product Design and Development*. McGraw-Hill, New York, NY, U.S.A.

Vacca, J. R. (1996) *VRML Bringing Virtual Reality to the Internel*, pp. 359-440. AP Professional, Boston, MA, U.S.A.

Van den Bout, D. E., J. N. Morris, D. Thomae , S. Labrozzi, S. Wingo, and D. Hallman (1992) Anyboard: an FPGA-BASED reconfigurable system. *IEEE Design & Test of Computers*, **39**(9), 21-30.

Van Leeuwen, E. H. and D. Norrie (1997) Holons and holarchies: intelligent manufacturing systems. *Manufacturing Engineer*, **76**(2), 86 -88.

Von Hipple, E. (1988) *The Source of Innovation*. Oxford University Press, New York, NY, U.S.A.

Ward, W., J. F. Liker, J. J. Cristiano, and D. K. Sobek (1995) The second Toyota paradox: delaying decisions can make better cars faster. *Sloan Management Review*, **36**(3), 43-61.

Weston, R. H., R. Harrison, A. H. Booth, and P. R. Moore (1989) Universal machine control system primitives for modular distributor manipulator systems. *International Journal of Production Research*, **27**(3), 395-410.

Walker, I. J. (1993) Requirements of an object-oriented design method. *Software Engineering Journal*, **7**(2), 102-113.

Woolsey, J. P. (1994) 777. *Air Transport World*, **33**(1), 22-31.

Yourdon, E. and L. L. Constantine (1979) *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*. Prentice-Hall, Englewood Cliffs, NJ, U.S.A.

Overview of Modular Product Development

模組式產品發展之綜述

黃俊哲

國立暨南國際大學資訊管理研究所

摘　要

傳統的製造業正轉型成一種敏捷機動的製造形式(agile manufacturing)，而一個敏捷機動的公司應能合理且有效地應用製造設備、並及時地以低成本產出不同形式的產品。在敏捷機動的產業中，產品的模組化設計與可重構流程乃是重要的一環。產品的模組化設計是經由各個製造元件與模組組合而產出大量不同形式的產品，每種產品因而具備不同而獨特的功能、特性與效應。這種模組組合的生產方式，已成為注意的焦點並經常列為同步化工程設計的目標。然而在論文文獻上並未引起太多的注意，解決方案也多表現在觀念階段上。本篇論文覆閱了模組式產品發展相關論文，並規劃出研究主題以提供學術對此領域之探討。