A Standard Description of the Terms Module and Modularity for Systems Engineering

Mahmoud Efatmaneshnik, Shraga Shoval, and Li Qiao

***Abstract*—The terms module and modularity are not part of the technical taxonomy in any of the systems engineering standards, which do not regard a module as a part of the formal system breakdown structure. In this paper, we redefine the term module as a unit composed of a set of components with a set of specific interfaces. This unit serves one or more purely nonfunctional goals, such as flexibility, evolvability, manufacturability, testability, and maintainability. According to this definition, a configuration item, a subsystem, an assembly, a subassembly, or a component can all be regarded as modules as they can serve nonfunctional goals. The important assertion here is that a module's boundaries do not necessarily coincide with those dictated by the functional or spatial system decomposition and hierarchy. The aim of this paper is to lay the foundations for the future standardization of various engineering design processes based on modularity for nonfunctional benefits. A clear definition of the terms module and modularity can assist systems designers and developers to optimize the value of a modular system. This research highlights the present inconsistencies in the field of modular system design and puts forward some critical questions, which will shape the future research into this field.**

***Index Terms*—Module, modularity, nonfunctional requirements, system engineering (SE), SE standards, SE process, system ilities.**

I. Introduction

MODULE and modularity are frequently used by developers and users of engineering systems during various lifecycle stages of the system. The terms are widely used in the product engineering literature, often equated with other terms such as "subsystem," "component," and "subassembly." However, the terms "module" and "modularity" are not standard systems engineering (SE) terms. The system hierarchy description in the SE standards and handbooks, including SE INCOSE handbook, IEEE 1220, ISO/IEC 15288, SEBOK (System Engineering Body of Knowledge) and MIL-STD-499 do not refer to these terms at all. The SE-VOCAB (Software and Systems Engineering Vocabulary—https://pascal.computer.org/sev_display/search.action) returns the following definitions on a search on "module" as with a particular note:

1) program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading;
2) logically separable part of a program;
3) set of source code files under version control that can be manipulated together as one;
4) collection of both data and the routines that act on it.

The terms "module," "component," and "unit" are often used interchangeably or defined to be subelements of one another in different ways, depending upon the context. However, the relationship between these terms is not yet standardized. This paper aims to provide ontological descriptions for these terms, which will contribute to the establishment of standard SE procedures that can be used for creating modular architectures.

Laying the foundations for standardization of various engineering design procedures based on modularity will provide a unified description across multidisciplinary and interdisciplinary fields such as manufacturing [1], assembly [2], [3], and testing processes [4]. In practice, clear definition of the terms module and modularity can assist systems designers and developers to maximize the value of a modular system design and provide financial valuation for capital budgeting purposes.

The design and construction of systems according to the SE approach is commonly performed using the top–down approach. The definition of a system hierarchy according to the IEEE 1220 standard is shown in Fig. 1. A system is composed of products that are, in turn, composed of subsystems, which consist of assemblies and components. The components consist of subassemblies, subcomponents, and parts. Following this standard approach, the system's required functions are decomposed and derived from the system's goals and objectives; and the system's hierarchy is a natural result of a one-to-one allocation of the set of hierarchical requirements to the physical descriptions that satisfy those requirements. In principle, a healthy set of requirements is modeled by a perfect tree-like graph, where there are no links between its leaves.

Assignment of the functional requirements to the configuration items creates a natural hierarchical-modular setting along the functional boundaries. For example, the SE V model is based on decomposition of the functional (or logical) descriptions, assigning each of the functional requirements to a specific configuration item, followed by the integration process that includes testing and validation. This process naturally creates a modular system along the functional boundaries. However, the boundaries of the modules in a system do not always need to fol-

Manuscript received May 14, 2018; revised August 28, 2018; accepted October 24, 2018. Review of this manuscript was arranged by Department Editor C.M. Eckert. *(Corresponding author: Mahmoud Efatmaneshnik.)*

M. Efatmaneshnik and L. Qiao are with the School of Engineering and Information Technology, University of New South Wales, Canberra ACT 2612, Australia (e-mail: m.efatmaneshnik@adfa.edu.au; l.qiao@adfa.edu.au).

S. Shoval is with Ariel University, Ariel 40700, Israel (e-mail: shraga@ariel.ac.il).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEM.2018.2878589

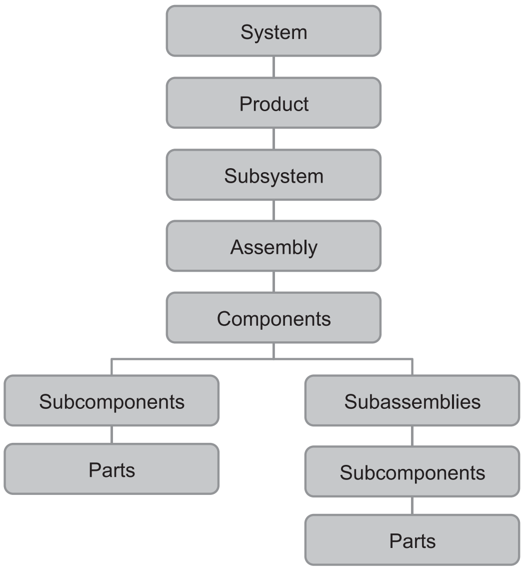

Fig. 1. System elements defined by functional boundaries (from IEEE 1220). The system's physical elements have one-to-one mapping to elements of functional decomposition or logical description of the system.

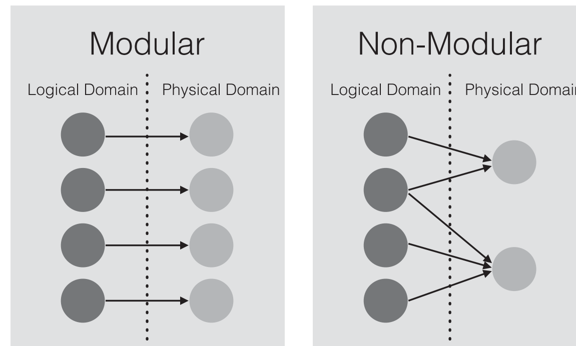

Fig. 2. Architectural view of modularity.

low the logical and functional boundaries. Furthermore, in some cases, a module's boundaries that follow the logical and functional boundaries may limit the system's performance during some of its lifecycle stages [5].

This paper is organized as follows. First, we look at common definitions of "module" and "modularity" in the product-engineering literature and in natural hierarchies of systems. We then summarize the outlined benefits of modularity and suggest a definition for the term "module." Following this defnition, we show that the boundaries of a module are not necessarily the boundaries defined by the functional units, namely, subsystems and components. Next, we discuss the effect of interfaces on modularity, and illustrate this with three standard products.

II. Common Notions for the Term "Module"

The word "module" comes from the Latin word "modulus," which means a "unit." Module as a term has found extensive usage in the product design community, however, for subtly different notations. Here, we review the typical usages of "module," which essentially define modularity as an engineering concept. We can categorize these definitions into two main groups of architectural and structural definitions. The system architecture is defined by the relationship between the functions and the components at a certain level of the functional decomposition hierarchy [6], or by the relationship between the models of logical and physical representations of systems [7]–[9]. The system structure, on the other hand, is defined by the relationship between the components [10], [11]. Fig. 2 depicts the notion of the architectural modularity: that is, the quality of having a one-to-one relationship between functions and components.

There are various methods with different characteristics to identify a system architecture. For instance, the functional structure heuristic methods are highly dependent on the designers' comprehension of the product [6]. The Modular Function Deployment, presented in [12], leaves module interaction choice to the designer, which is more management oriented [13]. In contrast, structure identification approaches based on the design structure matrix (DSM) are found to be more engineering oriented, repeatable, and computationally verifiable [14]. This is because the DSM is a common tool used to represent interactions among components in a product or a system. According to the specific application, the definition of the relationship between the components in the DSM is flexible. For instance, the interactions can be defined by spatial configuration, information, material, or energy exchange [15], as well as by their affinity, which is more important for assembly purposes [16].

Gershenson *et al*. [17] provide a comprehensive literature review on the use of the term "module" in engineering design. However, little classification of these definitions is made, except along specific domains such as software engineering, manufacturing. This paper suggests that there is consensus in some domains, but these definitions do not allow for the full benefits of product modularity (to be reached). They define a module as "a grouping of components that are similar in the life-cycle processes they undergo, and independent from all components outside of the module as they go through their lifecycle." Although we mostly agree with this definition, it is not precise enough to be used as a basis for standardized use in SE. Examining the literature, we see that the notion of the module has been used in variety of contexts with slightly different meanings.

A. *Functional Unit Designed Independently*

One of benefits of modularity is the ability to independently design, redesign, and evolve modules [18]–[21]. The authors in [18] emphasize that observance of the independence axiom in the axiomatic design theory necessarily leads to uncoupled functional requirements, which in turn facilitates the module creation. According to [21], a module is a unit within a product or a system that functions as an integrated whole but is designed independently. Baldwin and Clark [20] mention that modules are designed and produced independently. Modularity allows both design and production tasks to be divided among groups that can work independently and do not have to be part of the same firm/organization. In [22], the success of the computer industry is attributable to modularity, where a complex product or process is built from smaller subsystems that can be designed independently, yet function together as a whole.

B. Functional Unit With Strong Distinguishable Boundaries

This notion is mostly used for easier creation of product variants and product family platforms, by mixing and matching functional modules. The authors in [6], [23], and [24] use modules in this sense. Stone *et al.* [6] propose a function heuristic to identify modules within a product, and equipping them with standard interfaces. Huang and Kusiak [24] propose the insertion of the functional flow information into an interaction matrix, and then, clustering the matrix to identify the modules. Ethiraj and Levinthal [19] find that module recombination (or mixing and matching) plays a powerful role in the creation of adaptive designs. For module selection to be a useful mechanism of adaptation to new market demands, the modules need to be relatively fixed and interchangeable between design organizations. The fitness of the candidate modules depends, among other things, on the rigidness of their boundaries. Accordingly, Baldwin and Clark [20] suggest that new module designs may be easily substituted for old ones at a low cost to the system, which suggests that modules are always distinct parts of a larger system with strong distinguishable boundaries.

C. Functional Unit With Standardized Interfaces

Modular product architecture consists of modules, often clusters of physical components or functional building blocks with standardized interfaces [12], [25]. Borjesson and Hölttä-Otto [26] summarize that the standardized interfaces allow different sizes or versions of a module to be interchanged, enabling a timely response to the wide variety of demands from the consumer market. Huang [27] believes that the aim of modularity is to identify independent, standardized, or interchangeable units to satisfy a variety of functions.

D. Physically Detachable Functional Unit

In [28], the aim of a modular design is to develop a product architecture that consists of physically detachable units. The benefit is that the designs of previous modules can be used in a new module without any changes, or at least with minor variations, due to standardization of the functions and interfaces. Schilling [29] gives several examples of modularity in design, such as publishers who enable instructors to assemble their own text books from book chapters, and home-appliance manufacturers that offer their products in modular configurations, e.g., some stoves offer customers removable burners and plug-in barbecue grills and pancake griddles. The authors in [30] and [31] provide detailed methodologies for identifying components exposed to likely design changes, and modularization of those components to minimize the change propagation effects, thus limiting the redesign, remanufacture, and reimplementation costs of the changes systems.

E. Physically Detachable Cluster of Components

In [32], product modules are defined as subsystems within a product that are bundled as a unit, and serve identifiable functions. In addition, this paper denotes the difference between product modules and portfolio modules. The former refers to the pair comprising the subsystem and the functions; the latter are product modules that are used in multiple products. This kind of definition leaves much freedom in determining how a product family should be constructed. However, this is typical in large firms that can fully control the definition of a component cluster such as Volkswagen, which consists of several brands such as VW, Audi, Skoda, and Seat. Similarly [33] and [34] describe modules as products within a larger product.

F. Assembly as a Cluster of Components

In [35], the focus is on the aspect of product assembly and defines the module as product components that are assembled using a specific design architecture. Accordingly, a case study using an automobile body-in-white process is used to demonstrate the proposed modular architecting technique.

Following the aforementioned definitions, a few questions remain.

1) Does anything that is well encapsulated and isolated constitute a module?
2) Does anything with an interface constitute a module?
3) Does every module need an interface?
4) Is it necessary that this interface is standard (unchanging)?

The answers to these question can provide more than just terminological discussion. A standardization of the terms related to modularity can contribute to the various technical design procedures, especially in complex systems. Moreover, in the SE, a unit of functional architecture is either a component, a subsystem, or a system; and a unit of components for assembly is referred to as a subassembly or assembly. So, the question is whether there is room for the standard use of modularity in the SE. Standardization of the terms related to modularity is becoming more beneficial as Industry 4.0 methodologies replace the traditional concepts and standards. A key feature in Industry 4.0 is the modular structure of systems that consists of cyber-physical elements. The modular structure must provide interoperability, transparency, and decentralization of decisions by creating intelligent networks of physical resources and computational capabilities that are integrated and interconnected. The challenge is to construct highly modular structures is multivendor interoperability of automation technology [36]. This way, even complex systems become more flexible, allowing an easy plug-and-play integration or replacement of entities. The standard use of modularity can assist system engineers in the configuration of such complex systems.

In the following section, we list the benefits of modularity and note that all these benefits are nonfunctional.

III. Modularity Benefits

Decomposability is a fact of life; however, the ease of decomposability varies from system to system. In some systems, decomposability is natural, but in others, it might be very difficult to separate certain parts from the system. A module is often associated with the “ease of separation” from the rest of the system. However, questions may rise during the decomposition process, such as: Why do we want to create an artificial separation? At what cost?

Modularity creates a product platform that facilitates product variety, customization, and rapid upgradability throughout the system's lifecycle. Modularity generally makes it easier to maintain and evolve the system. The benefits of modularity are described extensively by [21] as the "power of modularity." However, modularity has limited capacity to adapt to the system's evolution, which we call "confined evolvability." For example, the benefits of modularity may not justify the required intellectual capital if the rate of changes of the market requirements is very slow. Similarly, modularity may not justify itself if the rate of change during the desired scope of the system's functionality in the target environmental context is too fast. In such an environment, the technological landscape will change too quickly. For example, consider portable memory storage devices such as 5 1/4' floppy drives, Zip drives, and miscellaneous hard drives, memory cards, and USB sticks. Each of these devices uses a different technology, so investment in their modularization might not lead to a return, or to incremental performance improvement.

Although there is a broad consensus in the SE about the definition and formats of functional requirements, the role of nonfunctional requirements is rarely discussed [37]. Glinz [38] discusses the definition, classification, and representation of nonfunctional requirements in the software engineering domain, and presents a list of 11 different definitions and references. The definitions include "nonbehavioral aspects of a system" [39], "overall attributes of the system" [40], requirements that are not specifically concerned with the functionality of a system [41], and a few more definitions. Here, we propose an additional aspect for nonfunctional requirements that are directly related to the system's modular architecture. Nonfunctional requirements usually have utilities for a specific system lifecycle, or for specific stakeholders. For example, the provision of modules to facilitate maintainability may be different from those designed to facilitate testability. The benefits of modularity include the following nonfunctional properties, also known as –ilities [33], in the various life-cycle stages.

1) Modularity in design:
   a) manageability of design process [42]–[44];
   b) evolvability or technology push (version creation) [20], [45];
   c) increased chance of innovativeness [46]–[48];
   d) style creation, customization, and modifiability [33], [49];
   e) changeability [30], [31].
2) Modularity in production and development:
   a) manufacturability [50]–[52];
   b) assembility [53], [54];
   c) ease of production planning [55]–[57];
   d) outsourcing [58]–[60];
   e) quality control [61];
   f) better supplier management [62]–[64];
   g) testability, verifiability (better test and verification) [65].
3) Modularity in operation:
   a) reusability [66], [67];
   b) flexibility [50], [68];
   c) serviceability and maintainability [69]–[71];
   d) upgradability [20], [45].
4) Modularity in retirement:
   a) recyclability [67], [72], [73];
   b) disassembility [74].

TABLE I
OUTLINES THE MAIN PARADIGMS OF SYSTEM OBJECT TAXONOMY

Object	Main paradigm
System of systems	Goal sharing Loose coordination
System	Function sharing Functional synchronization Integration Delivery of complex functionality
Subsystem	Simple function delivery Tight coordination High level of integrality
Component/part	Single function delivery Finds value in system context
Module	Delivery of system attributes/non-functional requirements
Sub-module	Delivery of a single system attribute/non-functional requirement

Modularity and functional requirements may not have positive correlation [75], [76]. However, modularity is frequently derived from nonfunctional requirements. Furthermore, modularity may have adverse immediate effects on functionality as it may involve handling of the functional structure for purposes other than pure functional goals [76].

IV. Proposed Definition for the Term "Module"

As shown, modularity often does not fit into the regular categorization of a system's functional or physical hierarchy, which shows a direct mapping of functional hierarchy and physical hierarchy. Here, we propose to use the term "module" as a preserved keyword indicating a segment of a system with a distinct boundary that relates to a particular *nonfunctional requirement(s)* of the system. Modularity should not be regarded as a driver for functionality and vice versa. In fact, it is often quite the opposite as modularity may have immediate negative consequences on functionality. However, in the longer term, it can, indirectly, improve functionality through facilitating nonfunctional attributes (e.g., through facilitating adaptive designs). Table I outlines the major differences between a module and other systemic objects such as the system itself.

Let us define "system parts" as "system atoms" that are the lowest possible decomposition that can be abstracted while still relevant to the functional or performance requirements lists. For example, a bicycle's spokes are elements of the bicycle because they have functional relevance to it, but the iron ore that the spokes are made of is not an element of the bicycle. The system's attributes and its boundaries determine the level of abstraction, as well as the limits across the physical and temporal dimensions by which the system elements can be described. We can now proceed to the formal definition of the term "module" in a systems context as follows.

A module is a group of some of the system's elements (components, subsystems, etc.), with a physical or notional boundary,

TABLE II
COMPARISON BETWEEN THE PROPOSED DEFINITION AND DEFINITIONS IN THE LITERATURE

Module definition	**Properties**				
	Non-functional utilities	Boundary	Physical detachability	Notional detachability	Standard interface
A functional unit designed independently	+	+	+	+++	+
A functional unit with strong distinguishable boundaries	++	+++	+	+	+
A functional unit with standardized interfaces	++	++	+	++	+++
A physically detachable functional unit	++	++	+++	+	+
A physically detachable cluster of components	+++	++	+++	++	++
The proposed definition	+++	+++	+++	+++	+

and is detachable, either physically or notionally from the system that, by this detachability alone, has a nonfunctional utility for a system lifecycle stage(s) or for a stakeholder(s)

This definition has the following fundamental elements.

1) Modules are groups of some system elements. This means that a subsystem can be grouped with some components from another subsystem to form a module.
2) Like the way the functional requirements are mapped into the subsystems, the nonfunctional requirements can be mapped into modules.
3) Modularity is the property of a system composed of modules.
4) A module should not necessarily have a unique function or some unique functions because the primary goal of modularization is nonfunctional.
5) Having unique functionalities (again from certain perspectives) should not be regarded as a criterion for being regarded as a module.
6) Subsystems and components only have functional purposes attached to them.
7) An assembly/subassembly is a module from a manufacturing perspective or viewpoint.
8) Every subsystem is a module. However, a module may or may not be a subsystem. In other words, the boundaries of the modules and subsystems are not necessarily identical.
9) A module does not necessarily have a standard interface. Having a standard interface should not be regarded as a criterion for a unit to be regarded as a module.
10) If a stakeholder is a common stakeholder across many systems, then the detachment mechanism might be enforced by a standard interface.

Table II draws comparison between the proposed definition and the definitions in the literature outlined in Section II. In this table, "+" indicates a weak relationship, "++" indicates a moderate relationship, and "+++" indicates a strong relationship. The score of each item in the table is determined based on the following three questions.

1) Can the property be derived from the definition?
2) Does the property clarify the definition?
3) Does the property necessarily relate to the definition?

The number of positive answers to these questions determines the grade. For example, the definition of a module as "A functional unit designed independently" does not necessarily relate to some or any of the nonfunctional utilities, the nonfunctional utilities cannot be directly derived from the independent design of the units, and independently designed units do not necessarily relate to the nonfunctional utilities (and therefore, the score of this item is "+"). Similarly, independent design does not determine clear boundaries of the system's elements, and boundaries do not necessarily clarify or be related to the design, again, resulting in a score of "+." On the other hand, the definition of a module as "a functional unit with strong distinguishable boundaries" can determine many nonfunctional utilities (as well as functional utilities), the nonfunctional utilities determine distinguishable boundaries and there are clear relations between the distinguishable boundaries and the nonfunctional utilities, resulting in a score of "+++" in the table.

The proposed definition does not necessarily disagree with any of the previous definitions but rather encompasses other definitions of a module, thus emphasizing a wider utility for the system design. Based on the definition for a module, the following definitions for modularity, modular, and modularization ensue noting that each of these can 1) an attribute of a system, or 2) an attribute of a unit within the system (e.g., a subsystem, a component, a cluster of components, etc.).

1) Modular.
   a) A modular system either has modules or has certain qualities that ease or make possible the process of modularization.
   b) A modular unit is either a module or a unit within the potential to be a module.
2) Modularity both a as a system and unit attribute is the quality of being or the potential to become a modular.
3) Modularization for both system and unit is the process of becoming modular.

The examples in Section V present in depth illustration of these definitions.

A. Mathematical Representation and Clarification of Some Issues

For completeness and clarity of the module definition, a set of theoretical representation follows. These equations support the proposed differentiation of modules and subsystems and also help to distinguish modularization and standardization. They also help in better understanding of the architectural and structural modularity. Consider system $S = \{S_P, S_F, S_{\mathrm{NF}}\}$ composed of *m* distinct subsystems described in physical description (S_P). S also has a functional (S_F) description and a nonfunctional ($S_{\rm NF}$) set of requirements

$$S_P = \{s_1, s_2, \ldots, s_{n_p}\}$$
$$S_F = \{f_1, f_2, \ldots, f_{n_f}\}$$
$$S_{\rm NF} = \{r_1, r_2, \ldots, r_{n_{\rm nf}}\}. \quad (1)$$

Such that

$$S_P \xrightarrow{\text{realizes}} S_F$$
$$S_P \xrightarrow{\text{realizes}} S_{\rm NF}. \quad (2)$$

The system *S* has n_p distinct subsystems (s_i) and each subsystem contains a number of physical components (e.g., mechanical, electrical, software components, and interfaces). The subsystem *i* has n_i components

$$s_i = \{c_i^1, c_i^2, \ldots, c_i^{n_i}\} \quad (3)$$

such that for any $1 \leq i \leq n$, we have

$$s_i \overset{\text{realizes}}{\rightarrow} \{f_{i_1}, f_{i_2}, \ldots, f_{i_m}\}, i_1, i_2, \ldots, i_m \in \{1 \ldots n_f\} \quad (4)$$

which means a set of main system functions are satisfied/delivered/realized by a subsystem. For representational simplicity, one layer of functional decomposition is considered. When for all subsystems $m = 1$, we say the system is architecturally modular meaning there is one to one correspondence between subsystems and main functional requirements (as depicted in Fig. 2). Function sharing takes place [78] when $i_m > 1$ and we say the system is architecturally nonmodular. A module *M*, on the other hand, is a set of components that satisfies/delivers/realizes at least one nonfunctional requirement

$$M = \cup c_i^j,$$
$$j \in \{1 \ldots n_i\} \text{ and } i \in \{1 \ldots n_p\} \quad (5)$$

such that

$$M \xrightarrow{\text{fully or partially realizes}} R_{\rm NF}, R_{\rm NF} \subseteq S_{\rm NF} \quad (6)$$

which reads module *M* satisfies a set of nonfunctional requirements $R_{\rm NF}$ as a subset of Systems *S* nonfunctional requirements set $S_{\rm NS}$. We can also define a submodule as a subset of *M* that fully or partially satisfy a single nonfunctional requirement r_i.

This mathematical representation helps to clarify a couple of issues. First, it cannot be ruled out that there might be a $j_{\rm th}$ component in the $i_{\rm th}$ subsystem (c_i^j) that also fully or partially contributes to the fulfillment of the nonfunctional requirement r_i, In the particular context of software engineering (as a subset of SE), this issue has been addressed extensively through the notions and practices of aspect-oriented programing (AOP). In AOP, aspects, which are roughly some combination of functional and nonfunctional requirements, can be mapped directly to pieces of the code, also known as modules. In this paper, the definition of a module is focused on the module's boundaries, whereas in AOP, it is also a question of components (codes in this case) that are created for the fulfillment of an aspect (that can be of functional or nonfunctional nature). The second issue is that some define the usage of standard components as modularity. This means that components c_i^j and c_l^k are the same type of components or are exactly the same component (which can happen in software where a piece of code is called/used by different parts of the program).

Obviously, standardization and modularization are not the same terms, but are often used synonymously in the product design literature, for example, in [20], [24], [29], [50], [68], and [78]. This is because standardization also reduces the cost of implementing the system's ∼ilities, such as (product) version creation or evolvability, and flexibility in use (e.g., through porting). As a result, there is often a tendency to equate modularity with standardized interfaces. For example, consider the popular Lego kits, which consist of building blocks (functional units) with very toned standard interfaces. According to our definition, the Lego blocks are not modules, and the ease of mixing and matching that is associated with them does not necessarily come from modularity. In this case, it comes from using standard interfaces. A standard interface does not really create a module, as it does not group components together. However, standardized interfaces of modules do indeed reduce the costs associated with evolvability, version creation, and the other benefits of modularity. Standardization and modularity can, however, be used together, and so we can say that

Standardization reduces the cost of modularization, and increases the benefits of modularity.

The modules can be equipped with standard interfaces for even greater evolvability, flexibility, testability, etc., at reduced cost. In some situations, modularization and standardization are performed simultaneously. For example, in Martin's paper about Design for Variety (DFV) [79], the goal of the team is to design the product platform architecture so that as much of the design as possible is standardized across generations. For the parts of the design that cannot be standardized, the team will modularize them. Two indices, the generational variety index and the coupling index, are used to decide where to standardize and/or modularize. Sered [80] also focused on the use of DFV and proposed a method called standardization and modularization driven by the process effort, where the aforementioned indices are employed to decide which component to be made standard or modular, with the major goal of minimizing overall process effort.

V. Examples

In this section, we illustrate different views of modular systems using three standard products: a passenger vehicle, a home food processor, and a small electronic eye security system. The aim of these examples is to analyze the modularity in terms of the proposed definitions. We evaluate the modular architecture of each system in terms of 11 fundamental properties that characterize the proposed definition, using a tabular format. The properties are represented by a question, and the score that each item in the table is assigned with characterizes the answer to the question. The "+" indicates a weak or no relevancy of the modular architecture to the property (value of 1), "++" represents a medium (value of 2), and "+++" represents a strong relevancy of the modular architecture to the property (value of 3). We then sum up the scores of each property to determine the most dominant modular properties of the analyzed product, and we also sum up the score of all the properties for each modular

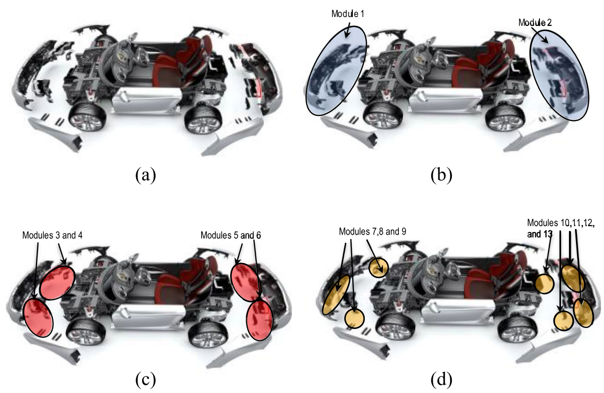

Fig. 3. Different modular views of a vehicle's chassis.

architecture. As stated previously, the horizontal summation indicates which are the principal properties in the product, while the vertical summation indicates which architectures fit with the proposed modular definition. The first two examples show how modules have different characteristics, some bearing more resemblance to a subsystem, and some having purely nonfunctional properties. This essentially is a hint that modularity of a unit is not a binary characteristic, rather it is a spectrum. The third example shows that the question of modularity can be also regarded as a characteristic of the system as a whole (again as a spectrum), and at this level, modularity is an architectural issue, rather than a purely structural problem.

A. Passenger Vehicle

We first consider a simplified decomposition of a vehicle shown in Fig. 3(a). The figure shows an exploded view of the major components that construct the vehicle. We chose this example as modern vehicles are basically a mobile network of mechanical, electrical, and electronic components. Examples of functional subsystems in a typical vehicle are the powertrain, braking, steering, suspension, cruise control, stability control, climate control, lighting, entertainment, communications, navigation, etc. Examples of nonfunctional requirement in a vehicle may be manufacturability, assembility, reusability, maintainability, etc.

The following figures illustrate how mobile modularization is updated in the various stages of the vehicle's lifecycle. Fig. 3(b) shows a modular view of the assembly process (modules 1 and 2). The two modules consist of many components of different subsystems with multiple functional characteristics, as well as other modules. This modular structure is beneficial during the assembly process, serving the nonfunctional requirement of *efficient robotic assembility*. Fig. 3(c) shows four modules (modules 3– 6) viewed during the maintenance stage. These modules construct the front and back lighting modules (right and left), and consist of both electrical (lights and wires) and mechanical components (casing and covers). These modules are interconnected with the electric control unit, and with the front and back modules of the chassis (respectively, modules 1 and 2). They serve nonfunctional requirements such as *fast and easy replacement*, *compatibility*, and *upgradability*. Finally, Fig. 3(d) shows a modular structure for the disposal stage (modules 7–13). Here,

TABLE III
MODULAR CHARACTERISTICS OF A VEHICLE

Properties	Modules			
	1,2	3-6	7-13	Total
1. Maps to non-functional requirements?	+++	+++	+++	9
2. Does not have a unique function?	+	++	++	5
3. Is not a subsystem?	+	++	+	4
4. Does not have standard interfaces?	++	+	+	4
5. Has a physical boundary?	+++	+++	++	8
6. Has a notional boundary?	+	+	+++	5
7. Helps in design?	+	++	+	4
8. Helps in manufacturing/assembly?	+++	+	++	6
9. Helps in service/operation/maintenance?	++	+++	+	6
10. Helps in retirement?	++	+	+++	6
11. Helps in evolution/upgradability/flexibility?	+++	++	+	6
Total	22	21	20	

Fig. 4. Modular food processor with its subsystems, components, and modules.

the modularization is based on the materials the components are made of. These modules serve nonfunctional requirements such as *reusability* and *recyclability*.

Table III presents the modular characteristics of the different modules during some stages in the vehicle's lifecycle.

Since the modular architecture of a vehicle is dynamic along the different stages of the lifecycle, we present the summation of values at each stage. As shown, the strongest relationship is with mapping the modules into nonfunctional requirements, followed by the existence of physical boundaries. These top scoring relationships fit extremely well with our formal definition of the term "module" in a systems context.

B. Home Food Processor

The next example is the food processor shown in Fig. 4 that consists of two subsystems, two modules, and seven components. Here, we differentiate between a subsystem (S), a component (C), and a module (M) by the functional/nonfunctional

TABLE IV
MODULAR CHARACTERISTICS OF A FOOD PROCESSOR

Properties	**Modules**				
	S1	S2	M1	M2	Total
1. Maps to non-functional requirements?	+	++	+++	+++	9
2. Does not have a unique function?	+	+	++	++	6
3. Is not a subsystem?	+	+	++	++	6
4. Does not have standard interfaces?	+	+	+	+	4
5. Has a physical boundary?	+++	+++	+++	+++	12
6. Has a notional boundary?	+	+	+	+	4
7. Helps in design?	+++	++	++	++	9
8. Helps in manufacturing/assembly?	+	+	+	+	4
9. Helps in service/operation/maintenance?	+++	+++	+++	+++	12
10. Helps in retirement?	+	++	++	++	7
11. Helps in evolution/upgradability/flexibility?	++	++	++	++	8
Total	18	19	22	22	

attributes of the items. The major subsystem (S1) provides several functional requirements such as power supply, user interface, attach/detach mechanism to the working counter, etc. Subsystem S2 provides other functional requirements such as mixing/grinding/mashing of liquids and solids. The seven components provide a specific functional requirement each (e.g., fine/coarse grinding, fine/coarse mixing, etc.). The two modules, on the other hand, provide nonfunctional attributes. For example, modules M1 and M2, which consist of several components, are designed for easy handling of liquids, and convenient cleaning. Notice that subsystem S2 can also mix liquids, but M1 adds the nonfunctional requirement of ease of handling and cleaning, and therefore, is considered a module. Notice that, in this system, the interfaces between the modules and subsystem S1 (as well as between S1 and S2) are standard, allowing quick and easy attachment/ detachment of the modules from the rest of the system. Such an interface also allows the user to add new modules as they become available, and possibly to combine modules from other systems (providing they adhere to the standard interfaces).

Table IV summarizes the modular characteristics of the modules (M1 and M2) and the subsystems (S1 and S2) of the food processor (we do not consider the components C1–C7 in this analysis as they are irrelevant to the modular characteristics). As in Table III, we sum up the scores of the modules and the subsystems according to their relations to the proposed modular fundamental properties. Similar to the previous example, the strongest relationships are with mapping the modules into nonfunctional requirements, the existence of physical boundaries and the help in the design, service, operation, and maintenance. To analyze the differences between subsystems and modules, we performed a vertical summation of the properties relationships. Although there are similarities between the modules and the subsystems in some modular characteristics, the total score of the modules is larger than the scores of the subsystems (22 versus 18 and 19).

C. *Electronic Eye-Controlled Security System*

Designing a modular architecture (for software and hardware products) is a multifaceted, challenging, and complex task [81]. Seliger and Zettl [81] propose a novel software tool for generating various modular configurations, and assessing the value of each one according to the strategic targets of the stakeholders as well as possible product scenarios. Each configuration is assigned with a benefit value using integer linear programming and a systematic classification optimization for determining the optimal configuration. The software tool was implemented in the design of a mobile phone and was found to be useful in reducing the development time and costs. However, the main drivers of the software tool are the functional requirements, as specified by the stakeholders. According to our proposed definition of modularity, the benefits of a specific modular configuration should be derived by the nonfunctional features of the design, as well as by the additional costs of the modularity, compared with the essential costs derived by the functional requirements.

A proper modular design can increase the product suitability for life-cycle economy [82] by enabling economical modifications, maintenance, and adaptations. Lower development costs of product variants and reduced manufacturing costs due to larger batch size production improve the product economy [83], particularly, in dynamic markets and diverse customer needs. However, modularity introduces some economic risks. These risks are due to the effort required in the development of the modular architecture, as well as the modifications to the initial nonmodular architecture. Since, according to our definition, modularity adds to the nonfunctional capabilities of the system, it must economically justify the additional investment. As an example, consider the electronic diagram of a simple electronic eye-controlled security system [84] shown in Fig. 5(a). The system consists of 17 units, each connected to 2 or 3 other units. The system can be decomposed into the following three independent functional subunits as shown in Fig. 5(b): sensing, signal processing, and response units. In this case, the system consists of 19 units and two interfaces (the additional two units are the power supplies for the two additional subunits). Alternatively, the system can be decomposed into four functional units, as shown in Fig. 5(c), with a single power supply and five intermodule interfaces. However, Fig. 5(d) shows a modular decomposition where nonfunctional features determine the clustering of the components. In the modular architecture shown in Fig. 5(d), the sensor, the buzzer, and the LED indicator are clustered in module 1, while all other low-cost electronic components (e.g., resistors, capacitors, transistors, and diodes) are in module 2 (module 3 can provide power supply to both modules or can be integrated into the modules). Such a modularization enhances reusability of the more expensive components in module 1, as well as evolvability of both the user interface components in module 1, and the electronic circuit design and production in module 2. A comparison of the three possible architectures is shown in Table V. As with the previous examples, architecture scores the highest

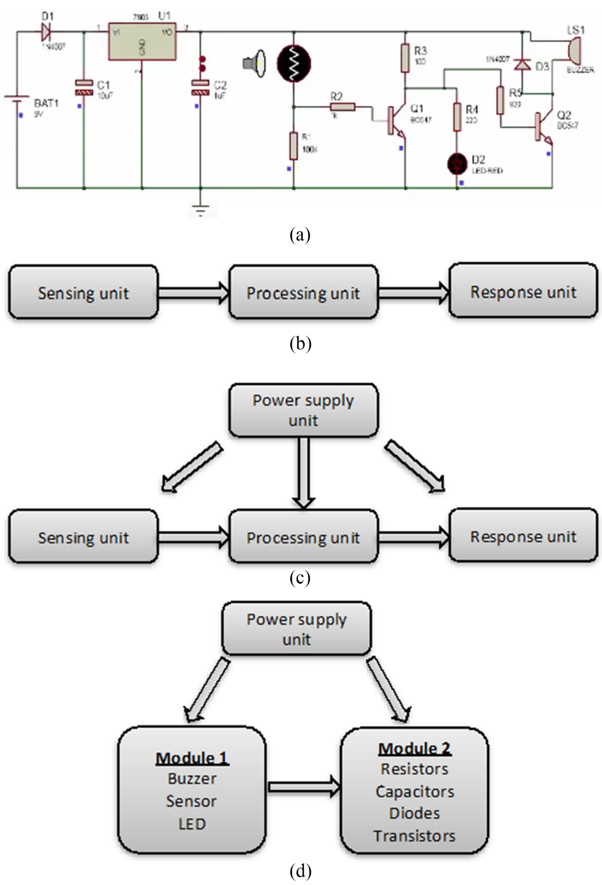

Fig. 5. (a) Electronic diagram of the eye-controlled security system. (b) and (c) Possible functional decompositions and (d) nonfunctional modular architectures.

TABLE V
MODULAR CHARACTERISTICS OF THE SECURITY SYSTEM

Properties	**Architectures**				
	a	b	c	d	Total
1. Maps to non-functional requirements?	+	+	+	+++	6
2. Does not have a unique function?	+	+	++	++	6
3. Is not a subsystem?	+	+	+	+++	6
4. Does not have standard interfaces?	+	+	+	+	4
5. Has a physical boundary?	+	++	++	++	7
6. Has a notional boundary?	++	++	++	+	7
7. Helps in design?	++	+	++	+++	8
8. Helps in manufacturing/assembly?	+	++	++	+++	8
9. Helps in service/operation/maintenance?	+	++	++	+++	8
10. Helps in retirement?	+	+	+	++	5
11. Helps in evolution/upgradability/flexibility?	+	+	++	+++	7
Total	13	15	18	26	

value in terms of the modular characteristics, mainly due to its nonfunctional attribute and the advantage during the design, manufacturing, operation, and other stages in its lifecycle.

VI. Conclusion

A module is a construct with nonfunctional goals within a system. Unlike a module, the "system," "subsystem," and "system elements" have functional goals and must contribute to the functional requirements. Naturally, it must have interfaces with other modules and subsystems, and be encapsulated by a boundary. The interfaces can be standard, but this is not essential. Standard interfaces are useful for the creation of product/system platforms as they allow repeated use of the same architecture. However, there are advantages associated with nonstandard interfaces, such as recyclability, maintainability, reusability, and adaptability. We can define a function as a value-delivering process that involves the transfer of material, information, and/or energy. Modules do not necessarily contribute to the enhancement of functions. Modularity may even reduce the performance. The modular system architecture is usually more complex than a monolithic/nonmodular one due to the additional interfaces and redundancies that modular architecture might require. Modularity usually creates a tradeoff between achieving nonfunctional attributes and functional performance. To conclude, we can state the following points regarding an ontological description of a module.

1) A module should not necessarily have a unique function or some unique functions because the primary goal of modularization is nonfunctional. Having unique functionalities (again from certain perspectives), should not be regarded as a criterion for being regarded as a module.
2) Every subsystem is a module. However, a module is not necessarily a subsystem.
3) An assembly/subassembly is a module from the manufacturing perspective or viewpoint.
4) A module does not necessarily have a standard interface. Having a standard interface should not be regarded as a criterion for a unit to be regarded as a module.

The importance of creating controlled usage for the term "module" is that it opens the way for creating standard processes for module creation, and formal processes for the inclusion of nonfunctional requirements in the design process. Such processes exist for functional requirements in IEEE 15288 and IEEE 1220 standards; however, for nonfunctional requirements, these standards do not include significant inputs. The vision we are promoting in this paper is that standard processes for categories of nonfunctional requirements, which are facilitated by modularization, are included in the formal design processes and their associated standards. In the next step of this study, we will rigorously identify the nonfunctional requirements associated with modularity and categorize them into groups based on the similarity of their corresponding modules and identify the type of required module boundaries (hard boundaries, or soft boundaries) for each requirements category. Another important challenge is the placement and integration of the nonfunctional design process with the functional design process. In a future work, we will advise on the techniques of module formation, its process, and its relationship with the functional design process.

Modularity is usually an afterthought to the functional design, and commonly various sorts of analysis are performed to determine whether modularization is a worthwhile practice. Gamba and Fusari [85] provide a valuation approach to the six modular operators proposed by Baldwin and Clark [21]. These operators consist of splitting (creation of a set of independent groups of components), substitution (changing of an existing module with a different one), augmenting (creation of new hierarchical levels), excluding (creation of a minimal system that can be incremented later), inversion (isolating the common features embedded in different modules), and porting (creation of a set of components compatible with other designs). They illustrate their valuation model in an example from the automotive industry where two manufacturers that produce similar types of vehicles, improve efficiency by merging design and production of some their components, by splitting their production processes, and by inverting the production of common components. These operators are considered to be imposed on an accomplished functional design, an assumption that implicitly considers modularity subsequent to the functional design. The product design community uses various sorts of often network-based modularity measures [86]–[88] to determine the potential of finished designs and established products for becoming modular. This trend is an indicator of the community's attitude, and implicit and explicit assumptions about the priority of functional requirements over nonfunctional ones.

This paper suggests that functional design and nonfunctional design need to be carried out together and are best tackled as an architectural issue. The limitation of this paper is in the lack of discussion on a methodology as to how this important issue can be tackled.

References

[1] S. Shoval, M. Efatmaneshnik, and M. J. Ryan, "Probabilistic approach to modular assembly," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5688–5693, 2017.

[2] S. Shoval, M. Efatmaneshnik, and M. J. Ryan, "Assembly sequence planning for processes with heterogeneous reliabilities," *Int. J. Prod. Res.*, vol. 55, no. 10, pp. 2806–2828, 2017.

[3] M. Efatmaneshnik and M. J. Ryan, "On optimal modularity for system construction," *Complexity*, vol. 21, no. 5, pp. 176–189, 2016.

[4] M. Efatmaneshnik and M. Ryan, "A study of relationship between system testability and modularity," *Insight Mag.*, vol. 20, no. 1, pp. 20–24, 2017.

[5] S. Shoval, "Dynamic modularization throughout system lifecycle using multilayer design structure matrices," *Procedia CIRP*, vol. 40, pp. 85–90, 2016.

[6] R. B. Stone, K. L. Wood, and R. H. Crawford, "A heuristic method for identifying modules for product architectures," *Des. Stud.*, vol. 21, no. 1, pp. 5–31, 2000.

[7] P. B. Kruchten, "The 4 + 1 view model of architecture," *IEEE Softw.*, vol. 12, no. 6, pp. 42–50, Nov. 1995.

[8] N. H. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*. New York, NY, USA: Addison-Wesley, 1985.

[9] I. Faulconbridge and M. J. Ryan, *Systems Engineering Practice*. Canberra, Australia: Argos Press, 2014.

[10] J. A. Zachman, "A framework for information systems architecture," *IBM Syst. J.*, vol. 26, no. 3, pp. 276–292, 1987.

[11] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems: A review and new directions," *IEEE Trans. Eng. Manage.*, vol. 48, no. 3, pp. 292–306, Aug. 2001.

[12] A. Ericsson and G. Erixon, *Controlling Design Variants: Modular Product Platforms*. Dearborn, MI, USA: Society of Manufacturing Engineers, 1999.

[13] K. Hölttä and M. P. Salonen, "Comparing three different modularity methods," in *Proc. ASME 15th Int. Conf. Des. Theory Methodol.*, 2003, pp. 533–541.

[14] L. Qiao, M. Efatmaneshnik, M. Ryan, and S. Shoval, "Product modular analysis with design structure matrix using a hybrid approach based on MDS and clustering," *J. Eng. Des.*, vol. 28, no. 6, pp. 433–456, 2017.

[15] S. D. Eppinger and T. R. Browning, *Design Structure Matrix Methods and Applications*. Cambridge, MA, USA: MIT Press, 2012.

[16] S. Shoval, L. Qiao, M. Efatmaneshnik, and M. Ryan, "Dynamic modular architecture for product lifecycle," *Procedia CIRP*, vol. 48, pp. 271–276, 2016.

[17] J. K. Gershenson, G. J. Prasad, and Y. Zhang, "Product modularity: Definitions and benefits," *J. Eng. Des.*, vol. 14, no. 3, pp. 295–313, Sep. 1, 2003.

[18] W. Chen, D. Rosen, J. K. Allen, and F. Mistree, "Modularity and the independence of functional requirements in designing complex systems," *Concurrent Product Des.*, vol. 74, pp. 31–38, 1994.

[19] S. K. Ethiraj and D. Levinthal, "Modularity and innovation in complex systems," *Manage. Sci.*, vol. 50, no. 2, pp. 159–173, 2004.

[20] C. Baldwin and K. Clark, "Modularity in the design of complex engineering systems," in *Complex Engineered Systems (Understanding Complex Systems)*, D. Braha, A. A. Minai, and Y. Bar-Yam, Eds. Berlin, Germany: Springer, 2006, pp. 175–205.

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

[22] C. Y. Baldwin and K. B. Clark, "Managing in an age of modularity," Harvard business review, vol. 75, no. 5, pp. 84–93, 1997.

[23] K. Ulrich, "Fundamentals of product modularity," in *Management of Design: Engineering and Management Perspectives*, S. Dasu and C. Eastman, Eds. Dordrecht, Netherlands: Springer, 1994, pp. 219–231.

[24] C.-C. Huang and A. Kusiak, "Modularity in design of products and systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 28, no. 1, pp. 66–77, Jan. 1998.

[25] S. Eppinger and K. Ulrich, *Product Design and Development*. New York, NY, USA: McGraw-Hill, 2015.

[26] F. Borjesson and K. Hölttä-Otto, "A module generation algorithm for product architecture based on component interactions and strategic drivers," *Res. Eng. Des.*, vol. 25, no. 1, pp. 31–51, 2014.

[27] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining Knowl. Discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[28] P. Gu and S. Sosale, "Product modularization for life cycle engineering," *Robot. Comput.-Integr. Manuf.*, vol. 15, no. 5, pp. 387–401, Oct. 1999.

[29] M. A. Schilling, "Toward a general modular systems theory and its application to interfirm product modularity," *Acad. Manage. Rev.*, vol. 25, no. 2, pp. 312–334, 2000.

[30] E. C. Koh, N. H. Caldwell, and P. J. Clarkson, "A technique to assess the changeability of complex engineering systems," *J. Eng. Des.*, vol. 24, no. 7, pp. 477–498, 2013.

[31] E. C. Koh, A. Förg, M. Kreimeyer, and M. Lienkamp, "Using engineering change forecast to prioritise component modularisation," *Res. Eng. Des.*, vol. 26, no. 4, pp. 337–353, 2015.

[32] J. B. Dahmus, J. P. Gonzalez-Zugasti, and K. N. Otto, "Modular product architecture," *Des. Stud.*, vol. 22, no. 5, pp. 409–424, 2001.

[33] G. Erixon, "Modular function deployment - a method for product modularisation," Ph.D. dissertation, KTH, Royal Inst. Tech., Sweden, 1998.

[34] G. Erixon, A. von Yxkull, and A. Arnstroem, "Modularity—The basis for product and factory reengineering," *CIRP Ann., Manuf. Technol.*, vol. 45, no. 1, pp. 1–6, 1996.

[35] T. AlGeddawy and H. ElMaraghy, "Optimum granularity level of modular product design architecture," *CIRP Ann., Manuf. Technol.*, vol. 62, no. 1, pp. 151–154, 2013.

[36] S. Weyer, M. Schmitt, M. Ohmer, and D. Gorecky, "Towards Industry 4.0-Standardization as the crucial challenge for highly modular, multi-vendor production systems," *IFAC-Papersonline*, vol. 48, no. 3, pp. 579–584, 2015.

[37] J. Mylopoulos, L. Chung, and B. Nixon, "Representing and using nonfunctional requirements: A process-oriented approach," *IEEE Trans. Softw. Eng.*, vol. 18, no. 6, pp. 483–497, Jun. 1992.

[38] M. Glinz, "On non-functional requirements," in *Proc. IEEE 15th Int. Requirements Eng. Conf.*, Delhi, India, 2007, pp. 21–26.

[39] A. I. Antón, "Goal identification and refinement in the specification of information systems," Ph.D. dissertation, Dept. Comp. Sci., Georgia Inst. Technol., Atlanta, GA, USA, 1997.

[40] A. M. Davis, *Software Requirements: Objects, Functions, and States*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[41] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*. Hoboken, NJ, USA: Wiley, 1998.

[42] M. Sako and F. Murray, "Modules in design, production and use: Implications for the global auto industry," presented at Int. Motor Vehicle Prog. (IMVP) Annu. Sponsors Meeting, 5-7 October 1999, Cambridge, MA, USA, 1999.

[43] M. Efatmaneshnik and C. Reidsema, "IMMUNE: A collaborating environment for complex system design," *Comput. Intell.*, vol. 1, pp. 275–320, 2009.

[44] M. Efatmaneshnik, C. Reidsema, J. Marczyk, and A. T. Balaei, "Immune decomposition and decomposability analysis of complex design problems with a graph theoretic complexity measure," in *Smart Information and Knowledge Management*. Berlin, Germany: Springer, 2010, pp. 27–52.

[45] M. Efatmaneshnik and R. Nilchiani, "Evolvability and upgradability of fractionated space system: A real options approach," in *Proc. AIAA SPACE 2012 Conf. Expo.*, 2012, pp. 5231–5232.

[46] P. Galvin and A. Morkel, "The effect of product modularity on industry structure: The case of the world bicycle industry," *Ind. Innov.*, vol. 8, no. 1, pp. 31–47, 2001.

[47] R. N. Langlois and P. L. Robertson, "Networks and innovation in a modular system: Lessons from the microcomputer and stereo component industries," *Res. Policy*, vol. 21, no. 4, pp. 297–313, 1992.

[48] J. Lee and F. M. Veloso, "Interfirm innovation under uncertainty: Empirical evidence for strategic knowledge partitioning∗," *J. Product Innov. Manage.*, vol. 25, no. 5, pp. 418–435, 2008.

[49] W. Y. Zhang, S. Y. Tor, and G. A. Britton, "Managing modularity in product family design with functional modeling," (in English), *Int. J. Adv. Manuf. Technol.*, vol. 30, no. 7-8, pp. 579–588, Oct. 1, 2006.

[50] A. Kusiak, "Integrated product and process design: a modularity perspective," *J. Eng. Des.*, vol. 13, no. 3, pp. 223–231, 2002.

[51] J. K. Gershenson and G. J. Prasad, "Modularity in product design for manufacturability," *Int. J. Agile Manuf.*, vol. 1, no. 1, pp. 99–110, 1997.

[52] S.-J. Park, F.-S. Kang, S. E. Cho, C.-J. Moon, and H.-K. Nam, "A novel switching strategy for improving modularity and manufacturability of cascaded-transformer-based multilevel inverters," *Elect. Power Syst. Res.*, vol. 74, no. 3, pp. 409–416, 2005.

[53] J. Paralikas, A. Fysikopoulos, J. Pandremenos, and G. Chryssolouris, "Product modularity and assembly systems: An automotive case study," *CIRP Ann., Manuf. Technol.*, vol. 60, no. 1, pp. 165–168, 2011.

[54] S. K. Fixson, "The multiple faces of modularity—A literature analysis of a product concept for assembled hardware products," Univ. Michigan, Ann Arbor, MI, USA, Tech. Rep. 03-05, vol. 1001, p. 48109, 2003.

[55] M. Efatmaneshnik and M. Ryan, "Balancing modularity: The role of uncertainty," in *Proc. Syst. Eng. Test Eval.*, Adelaide, Australia, 2014, pp. 1–12.

[56] M. Efatmaneshnik and M. J. Ryan, "On optimal modularity for system construction," *Complexity*, vol. 21, no. 5, pp. 176–189, 2016.

[57] S. Shoval, M. Efatmaneshnik, and M. J. Ryan, "Assembly sequence planning for processes with heterogeneous reliabilities," *Int. J. Prod. Res.*, vol. 55, no. 10, pp. 2806–2828, 2017.

[58] D. Campagnolo and A. Camuffo, "The concept of modularity in management studies: A literature review," *Int. J. Manage. Rev.*, vol. 12, no. 3, pp. 259–283, 2010.

[59] S. K. Fixson, Y. Ro, and J. K. Liker, "Modularisation and outsourcing: who drives whom? A study of generational sequences in the US automotive cockpit industry," *Int. J. Auto. Technol. Manage.*, vol. 5, no. 2, pp. 166–183, 2005.

[60] M. Sako, "Modularity and outsourcing," in *The Business of Systems Integration*, A. Prencipe, A. Davies, and M. Hobday, Eds. Oxford, U.K.: Oxford Univ. Press, 2003, pp. 229–253.

[61] B. Agard and S. Bassetto, "Modular design of product families for quality and cost," *Int. J. Prod. Res.*, vol. 51, no. 6, pp. 1648–1667, 2013.

[62] J. H. Mikkola, "Modularity, component outsourcing, and inter-firm learning," *R&D Manage.*, vol. 33, no. 4, pp. 439–454, 2003.

[63] M. C. Cooper and L. M. Ellram, "Characteristics of supply chain management and the implications for purchasing and logistics strategy," *Int. J. Logistics Manage.*, vol. 4, no. 2, pp. 13–24, 1993.

[64] R. I. Van Hoek and H. A. Weken, "The impact of modular production on the dynamics of supply chains," *Int. J. Logistics Manage.*, vol. 9, no. 2, pp. 35–50, 1998.

[65] M. Efatmaneshnik and M. J. Ryan, "A study of relationship Between of system testability and modularity," in *Proc. INCOSE Int. Symp.*, Edinburgh, U.K., 2016, pp. 1922–1931.

[66] H. El Haouzi, A. Thomas, and J.-F. Pétin, "Contribution to reusability and modularity of manufacturing systems simulation models: Application to distributed control simulation within DFT context," *Int. J. Prod. Econ.*, vol. 112, no. 1, pp. 48–61, 2008.

[67] Y. Zhang and J. Gershenson, "Questioning the direct relationship between product modularity and retirement cost," *J. Sustain. Product Des.*, vol. 2, no. 1-2, pp. 53–68, 2002.

[68] R. Sanchez and J. T. Mahoney, "Modularity, flexibility, and knowledge management in product and organization design," *Strategic Manage. J.*, vol. 17, no. S2, pp. 63–76, 1996.

[69] A. B. Abdullah, M. Yusoff, and Z. M. Ripin, "Integration of design for modularity and design for assembly to enhance product maintainability," in *Proc. 1st Int. Conf. 7th AUN/SEED-Net Fieldwise Semin. Manuf. Mater. Process.*, 2006, pp. 263–269.

[70] W. ElMaraghy and K. Meselhy, "Quality and maintainability frameworks for changeable and reconfigurable manufacturing," in *Changeable and Reconfigurable Manufacturing Systems*. London, U.K.: Springer, 2009, pp. 321–336.

[71] B. Nepal, L. Monplaisir, and N. Singh, "A framework to integrate design for reliability and maintainability in modular product design," *Int. J. Product Develop.*, vol. 4, no. 5, pp. 459–484, 2007.

[72] T. C. Kuo, H. C. Zhang, and S. H. Huang, "Disassembly analysis for electromechanical products: A graph-based heuristic approach," *Int. J. Prod. Res.*, vol. 38, no. 5, pp. 993–1007, 2000.

[73] H.-E. Tseng, C.-C. Chang, and C.-J. Cheng, "Disassembly-oriented assessment methodology for product modularity," *Int. J. Prod. Res.*, vol. 48, no. 14, pp. 4297–4320, 2010.

[74] C.-C. Huang, W.-Y. Liang, H.-F. Chuang, and Z.-Y. Chang, "A novel approach to product modularity and product disassembly with the consideration of 3R-abilities," *Comput. Ind. Eng.*, vol. 62, no. 1, pp. 96–107, 2012.

[75] A. C. Stryker and D. R. Jacques, "Modularity versus functionality—A survey and application " in *Proc. 7th Annu. Conf. Syst. Eng. Res.*, Apr. 20–23, 2009.

[76] K. Holtta, E. S. Suh, and O. D. Weck, "Tradeoff between modularity and performance for engineered systems and products," in *Proc. 15th Int. Conf. Eng. Des., Eng. Des. Global Economy*, 2005, pp. 2820–2833.

[77] K. T. Ulrich and W. P. Seering, "Function sharing in mechanical design," *Des. Stud.*, vol. 11, no. 4, pp. 223–234, 1990.

[78] A. Agrawal, "Product networks, component modularity and sourcing," *J. Technol. Manage. Innov.*, vol. 4, no. 1, pp. 59–81, 2009.

[79] M. V. Martin and K. Ishii, "Design for variety: Developing standardized and modularized product platform architectures," *Res. Eng. Des.*, vol. 13, no. 4, pp. 213–235, 2002.

[80] Y. Sered and Y. Reich, "Standardization and modularization driven by minimizing overall process effort," in *Proc. ASME Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, 2003, pp. 449–458.

[81] G. Seliger and M. Zettl, "Modularization as an enabler for cycle economy," *CIRP Ann., Manuf. Technol.*, vol. 57, no. 1, pp. 133–136, 2008.

[82] M. Hauschild, J. Jeswiet, and L. Alting, "From life cycle assessment to sustainable production: Status and perspectives," *CIRP Ann., Manuf. Technol.*, vol. 54, no. 2, pp. 1–21, 2005.

[83] G. Seliger, *Sustainability in Manufacturing*. Berlin, Germany: Springer, 2007.

[84] Electronics Hub, Electronic Eye Controlled Security System, May 2017. Available: http://www.electronicshub.org/electronic-eye-controlled-security-system/

[85] A. Gamba and N. Fusari, "Valuing modularity as a real option," *Manage. Sci.*, vol. 55, no. 11, pp. 1877–1896, 2009.

[86] J. K. Gershenson, G. J. Prasad, and Y. Zhang, "Product modularity: Measures and design methods," *J. Eng. Des.*, vol. 15, no. 1, pp. 33–51, Feb. 1, 2004.

[87] A. Cabigiosu and A. Camuffo, "Measuring modularity: Engineering and management effects of different approaches," *IEEE Trans. Eng. Manage.*, vol. 64, no. 1, pp. 103–114, Feb. 2017.

[88] S. Jung and T. W. Simpson, "New modularity indices for modularity assessment and clustering of product architecture," *J. Eng. Des.*, vol. 28, no. 1, pp. 1–22, 2017.

Authors' photographs and biographies not available at the time of publication.