Tolga Kurtoglu
Mission Critical Technologies,
NASA Ames Research Center,
MS 269-3 Moffett Field, CA 94035
e-mail: tolga.kurtoglu@nasa.gov

Matthew I. Campbell
Department of Mechanical Engineering,
The University of Texas at Austin,
Austin, TX 78712
e-mail: mc1@mail.utexas.edu

Cari Bryant Arnold
Engineering Design Program,
Penn State University,
University Park, PA 16802
e-mail: cari.arnold@psu.edu

Robert B. Stone
Department of Interdisciplinary Engineering,
Missouri University of Science and Technology,
Rolla, MO 65409
e-mail: rstone@mst.edu

Daniel A. Mcadams
Department of Mechanical Engineering,
Texas A&M University,
College Station, TX 7784
e-mail: dmcadams@tamu.edu

A Component Taxonomy as a Framework for Computational Design Synthesis

In this paper, we present our findings on the development of a taxonomy for electromechanical components. In building this taxonomy, we have two main objectives: First, we strive to establish a framework for future computational tools that archive, search, or reuse component knowledge during the conceptual phase of design. Second, we aim to define a standard vocabulary that derives uniformity and consistency in the representation of electromechanical component space. Through both empirically dissecting existing products and defining categories based on functional analysis, we defined 135 generic component types. The use and necessity of the resulting taxonomy by a suite of computational design tools are illustrated in two applications of conceptual design.
[DOI: 10.1115/1.3086032]

1 Introduction

Components are the fundamental artifacts from which physical devices are built. Their representation is essential to the successful conceptualization and realization of designed products. While formal component representations exist during the detailed stages of product development, electromechanical components lack similar representations that support the conceptual phase of design. The difficulty may hinge on the very nature of conceptual design, where the geometry of individual components and their dynamic behavior are still ill defined. At this stage of design, details of the components are less important than the ability to represent component knowledge at a higher level of abstraction. The *functionality* of components provides a natural framework upon which such abstractions can be built.

In this research, we are taking a first step for representing component knowledge from a functional perspective. Toward that goal, we are developing a *hierarchical taxonomy* for electromechanical components based on functional knowledge. The following four factors describe the motivation for the creation of this taxonomy.

Development of a computational framework for design synthesis. Many researchers have explored automated design tools to improve design synthesis activities [1–5]. Components constitute the fundamental building blocks of these activities. Within the variety of computer aided design research, various methodologies and tools have been developed, which require a rich set library of components. However, there is no agreement upon standard component library. As a result of this, libraries of components are independently developed in an application specific manner. One of the goals of this project is to reconcile these efforts and to arrive at an initial creation of an electromechanical component library that can be leveraged by a number of design automation methods. The use of this initial library by two computational tools is presented in Sec. 5. Both of these applications depend solely on the developed taxonomy to automatically create conceptual solutions to design problems.

Design knowledge reuse. Over the past few decades, systematic approaches to conceptual design have emerged [6–11]. These design methods begin by formulating the product function as a set of low level subfunctions, solutions to which are then synthesized together to arrive at a final design. The core of the computational synthesis methods [1–3] built upon this function-based framework is the mapping of subfunctions to components. This allows designers to generate concept variants from a generic functional description of the product being designed. All of these computational methods require a knowledge base of "reconfigurable" standardized component objects that can be archived, searched, and reused. The defined taxonomy facilitates the organization of such a knowledge base so that it can be used by various computational design tools. In Sec. 3.1, we show how the taxonomy is used to capture and organize component design knowledge by an online design repository. The two applications presented in this paper leverage information from this repository and follow the aforementioned function-based synthesis approach to automatically search for solution principles to given functional specifications. Reuse of existing design knowledge is the fundamental principle behind the development of these two tools.

Communication of design knowledge. The use of natural language leads to ambiguity in representing component design knowledge. Often, arbitrary and redundant component naming results in different interpretations among designers for similar concepts. These issues hinder effective communication of design knowledge. By associating fundamental component concepts with

Contributed by the Engineering Simulation & Visualization Committee of ASME for publication in the Journal of Computing and Information Science in Engineering. Manuscript received July 13, 2006; final manuscript received March 3, 2008; published online March 6, 2009. Review conducted by S. Gupta.

uniquely defined "component terms" and by providing definitions for each term, we strive to derive uniformity and consistency in the representation of components and to improve communication of design information both in industry and in design education.

Standardization for digital component cataloging. Solutions to conceptual design problems are usually represented as a configuration of components and interactions between them [3,12]. The transformation from these configurations to fully embodied design solutions requires the specification of a system of electromechanical components that meet the overall design requirements. Given the breadth of suppliers and production methods that exist today, most engineered artifacts are a mix of both custom-made parts and original equipment manufacturer (OEM) parts. As a result, the OEM suppliers compete by striving to improve their components' quality and variety. It is particularly important for them to catalog their solutions such that they can be efficiently retrieved and incorporated into the design process. Technologies involving electronic representations of standard components and resulting digital databases are becoming more prominent in engineering design [13–15]. Contributing to these efforts, our taxonomy provides a useful classification scheme for vendors selling a variety of OEM components.

Motivated by these factors, we provide a starting point for the creation of a component taxonomy that is accessible to all design engineers. The goal of this paper is not to capture all component types that physically exist but to create a function-based hierarchical framework of subcategories for all components to fit within. The resulting function-based hierarchical taxonomy provides a *novel* perspective in representing component design knowledge as follows.

- It provides a consistent means of classifying and otherwise defining new component concepts based on a component's "function."
- It allows one to capture functional relationships within and between components that is critical for the early stages of design where component geometry and dynamics are yet to be determined. Accordingly, the defined taxonomy provides a natural framework for representing component knowledge using varying degrees of functional abstractions. This is contrary to component representations that rely mostly on geometry information and component performance characteristics such as the ones found in design catalogs.
- It serves as a framework that bridges an important gap in the representation of component design knowledge and facilitates the development of a suite of computational design tools to be used for design synthesis.

2 Related Work

A widely used approach for cataloging knowledge about a certain domain is the development of ontologies. The concept of ontology, as used in knowledge engineering, is described as [16] "a term used to refer to the shared understanding of some domain interest." Neches et al. [17] stated "An ontology defines the basic terms and relations compromising the vocabulary of a topic area." Several ontologies have been proposed in the engineering design literature [12,18–22]. Liang et al. [12] suggested a port ontology as a tool for conceptual modeling and generating system architecture. Stahovich et al. [22] proposed that the fundamental ontology for mechanical devices should be based on physical behavior, not structure. This research attempts to determine the behavior of a mechanical device from a description of its structure and its deriving inputs. Horváth et al. [18,19] defined a general ontology to model design concepts and the interactions between them. The cataloged concepts are then initiated in a computer based functional synthesis process. Similar to this approach, Kitamura and Mizoguchi [20] proposed an ontology to describe functionality of physical artifacts based on what is referred to as "way of function achievement."

From a perspective different than ontology-based approaches, several researchers have developed systems to capture component functionality and interface requirements. The captured models range from detailed function and behavior information to lose hierarchical artifact relationships. Among those, the NIST Design Repository Project is a framework capable of storing component information and how the elements of information are related to each other [23]. Within the NIST model there are five sections: artifacts, functions, forms, behaviors, and flows, which contain information relative to their denoted name. The core product model (CPM) by Fenves et al. [24] is the embodiment of the NIST-driven representation efforts. Other function-based representations have been proposed by various researchers to support search and modeling processes of conceptual design [4,25,26].

The functional structure concept developed by Pahl and Beitz [7] provides a general and powerful representation of a design during conceptual stages. Researchers at University of Missouri-Rolla, University of Texas at Austin, and NIST have developed a standard vocabulary for describing the basic functional blocks for electromechanical design problems. This effort has culminated into a functional basis language that includes a set of terms that span the space of all functions and all flows [27]. Here, a function refers to a transformation operation from input flow to output flow. Functions are used in verb-object format. For example, a motor "converts electrical energy to mechanical energy." Three sets of function terms are defined to allow three levels of abstraction for allocating functions to a system.

In this research, we define a taxonomy for describing the electromechanical component space. Our taxonomy complements the functional basis representation [27] and allows the development of automated design tools such as Refs. [1–3,5] that would assist designers in mapping from functional requirements to component solutions and conceptual configurations.

3 Research Method

In this section, we present our approach for developing a comprehensive set of fundamental component concepts for electromechanical components. Our research includes two primary steps: an empirical analysis to study components of engineering artifacts, and a derivation and classification of component concepts and their definitions.

3.1 Function-Based Component Classification. A critical starting question for the presented research is to determine a systematic means of cataloging and classifying component design knowledge. Here, we take inspiration from one of the widely used tools employed in knowledge representation: lexicons. The most commonly used lexicon is the one developed by Chenhall [28] who stated the following.

"The lexicon … is based on the assumption that every man-made object was originally created to fulfill some function or purpose and, further, that original function is the only common denominator that is present in all of the artifacts of man, however simple or complex."

Given this definition, the AI community takes a similar approach to component classification by using the function and form of a component as fundamental elements in its classification. The presence of component function in component naming is an important linkage between the theory of knowledge capture and representation and the theory of design. An understanding of function is integral to the design process [7,11]; hence, a natural relationship between components and function must exist. In this research, we are exploring this relationship in defining a *consistent* classification scheme. Accordingly, we propose a means of capturing the latest knowledge of systems design using the Chenhall approach with a real time update. With this function-based per-

Fig. 1 Port analysis for an electric motor

spective, we defined a taxonomy for describing the electromechanical component space.

3.2 Empirical Product Analysis. The first step in our research is the dissection of consumer products. In this step, we analyze components within existing engineering artifacts. A set of consumer products is chosen for their low cost and wide variety, and design knowledge is extracted for all of the included components. In conducting the product dissections, we strictly follow the method presented in Ref. [11]. Accordingly, for each product analyzed, a bill of materials (BOM) is constructed. Each component is labeled and photographed as it is removed from the product and recorded in a tabular format listing its attributes.

This component design information is organized and stored in a web-based design knowledge repository[1] that is managed at the University of Missouri-Rolla. The design data are recorded into the repository using an open source cross-platform repository entry application. The entry template is organized based on individual components in a product. Accordingly, information about each component is recorded on separate templates, which are then aggregated to construct a product level representation. There are three types of attributes captured for *each component*: descriptive attributes, physical attributes, and functional attributes. Examples of descriptive attributes are part number, quantity, part description, hierarchical (assembly) information, and predicted manufacturing processes. Physical attributes include weight, type of material, and component specific parametric information (examples are height, width, length, inside/outside diameters, thickness, etc). These attributes are physically measured for each component and recorded into the data template. Finally, the functional attributes include the function(s) each component fulfills, the flows (energy, material, and signal) that pass through each component, and the connectivity and resulting interfaces of components. This information is captured by building a function structure for each product and then associating individual subfunctions of the functional model with components that are used to fulfill them.

In this study, over 100 consumer products are studied to arrive at the component taxonomy. Because of the level of detail captured for the online database, the empirical dissection of products and subsequent data entry into the repository is a very laborious process, typically ranging from 10–15 person-hours per product. These teardowns have been performed over the past 3 years through the efforts of several experienced graduate students and the occasional undergraduate research assistants. A comprehensive list of products that were analyzed can be found in Ref. [29].

[1]http://function.basiceng.umr.edu/repository.

3.3 The Classification Hierarchy. The goal of classification in this research is focused more on the practical use of the proposed hierarchical taxonomy. For this reason, we have chosen to initially begin with a function-based framework for the component classification hierarchy. The hierarchical framework was initially established from the notion that device function is an integral and critical characteristic of a component from the perspective of concept selection during the design process [7,11]. As a starting point, the lists of primary and secondary level function terms from the functional basis [27] were used to designate the primary and secondary levels of the component framework.

3.3.1 Establishing the Hierarchy. In order to successfully place a component term into the taxonomy, the functional traits of each component term need to be established. Here, a component is defined as having input and output ports through which it is connected to another component [20]. The functional traits of each component term are determined by analyzing the individual components housed within the repository of product information and categorized under that component term. The black box functionality for each component term is defined by identifying the most commonly occurring subfunction (function-flow combination) assigned to each of the components classified under that term in the repository.

In addition to primary functionality, the major flows through a component and the possible port connections to other components are also established for each component (see Fig. 1). When identifying port connections, the following port definitions are utilized.

(a) *Object port*. A device port through which a flow (material, energy, or signal) enters and then travels through the device from the input port to the output port and is processed by the device [20,27].
(b) *Medium port*. A device port through which a flow (material, energy, or signal) enters and then travels through the device from the input port to the output port while holding an object and enabling it to flow through the device (e.g., water can act as a medium carrying hydraulic energy as an object through a device) [20,27].
(c) *Assembly port*. A device port that acts only as a mating surface to support the weight or stabilize the position of the device.

Primary Component Classification	Secondary Component Classification	Component Term	Component Subset	Synonyms	Definition
Branchers	Separaters	...			
	Distributors	...			
Channelers	Importers/Exporters	...			
	Transferors	Carousel			A device used to move material in a continuous circular path.
		Conveyor			A device used to move material in a linear path.
		Electric Conductor		lead	A device used to transmit electrical energy from one component to another.
			Electric Wire		An electric conductor in the form of a thin, flexible thread or rod.
			Electric Plate		An electric conductor in the form of a thin, flat sheet or strip.
		Electric Socket			A device in the form of a receptacle that transmits electrical energy via a detachable connection with an electric plug.
		Electric Plug			A device in the form of a plug that transmits electrical energy via a detachable connection with an electric socket.
		Belt		strap, girdle, band, restraint	A device shaped as an endless loop of flexible material between two rotating shafts or pulleys used to transmit mechanical energy.
		...			
	Guiders	Hinge		pivot, axis, pin, hold down, jam, post, peg, dowel	A device that allows rigidly connected materials to rotate relative to each other about an axis, such as the revolution of a lid, valve, gate or door, etc.
		Diode			A semiconductor device which allows current to flow in only one direction.
		...			
Connectors	Couplers				
	Mixers				
...	...				

Fig. 2 An excerpt of component terms and definitions organized using the proposed hierarchical taxonomy

Flow information contained in the repository is used to identify all ports of a particular component. This information was then generalized across a component term group. For this research, the port definitions are qualitative in nature and do not reflect quantities of ports determined during embodiment phase of design (e.g., two assembly ports for a blade versus four). Additionally, since assembly ports are not used at this point to help classify a component term, these connections were ignored. Component term definitions within the hierarchical taxonomy were standardized using flow information from the port analysis in addition to common morphological characteristics of the components within a single group (see Fig. 2 for an excerpt).

The individual component terms and associated definitions represent the different "species" of components. The definitions of these terms are critical to the usefulness of the ontology proposed. In defining these component terms, disagreements may exist over how narrowly to define different terms. For example, should an axle and a drive shaft be classified under the same component term? Should a flexible hose be classified under a different component term than a rigid tube? In the case of the axle and drive shaft, these two components solve different functionalities and would, therefore, be placed under different branches of the proposed taxonomy. The flexible hose and rigid tube are functionally similar, so a decision must be made about whether to group them together under a broad definition or separate them into more specific groups. When defining terms, effort was made to determine whether a new (separate) definition would be beneficial from the perspective of a designer in the early conceptual stages of design, for example, deciding whether to use a flexible versus a rigid tube to *transfer* a *material* would be less useful when initially generating concepts than deciding whether to use a *tube* versus a *conveyor*. To help evaluate whether terms were defined at a low enough level of detail while preserving necessary coverage of the component space, additional consideration was made and two criteria were defined: *completeness* and *exclusivity*. The definitions for the two criteria are given below.

(a) *Completeness*. The measure of how well a list of components captures the complete set of all electromechanical components.
(b) *Exclusivity*. The measure of the independence of terms in the taxonomy. It is the opposite of overlap or redundancy of taxonomy terms.

These two criteria are enforced at the *component term* and *component subset* levels of the hierarchy (see Fig. 2 for an excerpt). The *completeness* criterion motivates one to construct a taxonomy that covers all the concepts that are relevant in the electromechanical component domain. In general, completeness within a domain is accomplished automatically with an empirical study of existing products. It is likely that with each additional product that is dissected and cataloged in the repository, the return in the number of newly defined fundamental component concepts begins to reduce. Of course, a large number of artifacts will need to be dissected before a change in the rate of new concepts is noticeably decreased.

The challenge in such research is to heed the *exclusivity* criterion. Technically, there could be as many component types as the number of components that exist in all electromechanical devices. This, of course, is impractical, and does not take advantage of component types that are well accepted (e.g., gears). As briefly described before, the key in establishing the exclusivity criterion is to carefully define what constitutes a new fundamental component type.

To achieve this, we follow two guidelines: (1) any fundamental component type should be as general as possible but specific enough to allow the user to build a clear abstraction of a component, which can be used during conceptual design. (2) The defined taxonomy as a whole should include a sufficient number of component terms (i.e., building blocks) to allow the user to represent a variety of concepts. By following these two guidelines, our efforts at defining an exclusive set of component concepts led to a total of 135 unique component terms. The complete list is published at the University of Missouri at Rolla Design Repository website [29].

4 Applications

In this section, we present two conceptual design applications that require a thorough online repository of component design knowledge. In these applications, the proposed component taxonomy permits the development of effective design automation tools. In each of the examples shown below, researchers utilize the functional representations contained in the online repository to enable a computer to determine what components are needed to instantiate a given functional description of a design problem.

Without a rigorously defined component taxonomy, these tools would be limited in usefulness and, in some cases, infeasible using reasonable computing resources.

4.1 Application I: A Graph-Grammar Framework for Automated Concept Generation. In the first application [3], we utilize the aforementioned design repository, and we derive design rules from it to capture the knowledge of the original designer's intent in mapping functional specifications to component solutions. In creating the rules, we observe the common uses of the components and formulate this knowledge as "grammar rules." A thoroughly developed component taxonomy is critical to the implementation of this approach, since the design patterns captured by the grammar rules are dependent on the component representation schema followed. Both completeness and exclusivity criteria become important as they determine the feasibility of capturing certain mapping patterns and the variety in the potential solution space. The functional subcategories of the developed taxonomy enable us to accurately capture the relationships between functional requirements and component concepts that are used to fulfill them. Moreover, without a standardized component representation, the grammar rule set grows unboundedly, deeming the development of the grammar infeasible.

This method is built upon a framework that represents design knowledge using graphs. In this application, two graph-based representations are utilized: function structures [7] and configuration flow graphs (CFGs) [30]. A function structure is a graphical representation of the decomposition of the overall function of a product into smaller, more elemental subfunctions. The subfunctions are connected by flows (specified as energy, material, or signal) that they operate on. Overall, a function structure represents the transformation of input flows into output flows at the product level. A CFG, on the other hand, is a graph representation of how functional components are connected. In a CFG, nodes of the graph represent product components, and arcs represent energy, material, or signal flows between components. For flow naming, the functional basis terminology is adopted, while the components of the graph are named using the component terms of the developed taxonomy. The CFG is a specific implementation of what some loosely define as the topology, the architecture, or the configuration of a product. The graph is also similar to an exploded view in that components (often drawn isometrically) are shown connected to one another through arcs or assembly paths. Figure 3 shows an example of a CFG (and a function structure) along with an exploded view for a camera product.

During product teardowns, we build an existing product's CFG and its function structure and capture the mapping between the graphs. Each mapping represents a *design decision* that shows how a functional requirement was transformed into an embodied solution in the actual design. In order to extract these mappings in a consistent manner, we utilize the flow information provided by the two graphs. We take advantage of the fact that the two graphs contain the same flow types. This allows us to "follow the flow paths" in both graphs and to define strict boundaries that isolate the mapping between functional nodes of a function structure and component nodes of a configuration flow graph. This procedure is illustrated schematically in Fig. 4, where three design rules derived from an empirical analysis of a product are shown. The grammar rules define what components can be used in creating a new design in order to meet certain functional requirements specified by the function structure of the new design. For example, the first rule states that if functions "store electrical energy" and "supply electrical energy" are required in a design problem, then a "battery" can be used to address the specified functional need. Similarly, a "driveshaft" and a "rotational coupler" pair can be used to "transfer rotational mechanical energy" (rule 2 in Fig. 4).

The grammar provides a method for automatically generating design configurations through a search-based execution of rules. This computational synthesis approach is to perform a graph transformation of the initial function structure of a product to be designed into a set of configuration flow graphs. The grammar affords a representation of the design space as a tree of solutions built from an initial function structure. Each transition down the tree adds more components to the design configuration, which incrementally builds to a final concept represented at one of the leaves of the tree. At the end, the search process returns different concepts with potentially varying degrees of complexity. (The details of the computational synthesis algorithm can be found in Ref. [30]). The grammar rules are stored in a database as part of the developed computational design tool[2] and construct a knowledge base of designer decisions that can be reused for generating conceptual solutions to given functional specifications.

Table 1 summarizes the results of designing a "bottle capping machine" using the graph-grammar based computational design approach. For this design problem, the grammar generates 339,168 unique design configurations that consist of a different selection and configuration of component concepts. As is shown in the last column of the table, these design configurations include 28 different component concepts from the developed taxonomy. (A more detailed report on the performance of the computational synthesis algorithm on various design problems can be found in Ref. [30].)

Moreover, the open endedness of the grammar formulation enables the synthesis of multiple-input multiple-output systems and accounts for function and structure sharing. Accordingly, the grammar rules developed are not simply one-to-one matches of functions to components. Instead, the grammar provides a more generic approach capable of mapping multiple components for a single function, or a single component for multiple functions, as is the case in function sharing, or multiple components for multiple functions. (Different mapping schemes of functions to components are shown in Fig. 4.)

4.2 Application II: Matrix-Based Concept Generation. The second approach also uses a function structure of a conceptual design as an input to generate feasible concept variants from a repository of design knowledge. In this method, detailed in a recent paper [2] as summarized in Fig. 5, matrix multiplication is used to translate user-input chains of subfunctions acting on flows into chains of components that both solve the desired functionality and exhibit compatibility. Function-component matrices (FCMs), defining the functionality of each component, and component-component matrices (commonly referred to as design structure matrices (DSMs)), defining the compatibility of component connections, are extracted from the previously mentioned online design repository and used to generate and filter an exhaustive set of design solutions.

The scheme begins with a functional model for either a new or redesigned product to be developed. Information extracted from a graphical block functional diagram created during a function-based design methodology and expressed using the functional basis terms described in Sec. 2 is used to seed the concept generation. The block diagram is translated into a matrix form that describes the adjacency between functions in the chain.

The next step utilizes the design knowledge gathered from existing consumer products to define relationships between a component and the functions that it solves. Reverse engineering techniques are applied to existing consumer products, and information extracted from each product's bill of materials and functional model is stored in the web-based design repository mentioned in Sec. 3.2. Information describing the functionality of each component is stored in the online database, and FCMs for individual products or specified groups of products can easily be generated from the stored information. Nonzero cell entries in the FCM indicate that the component from the column containing the cell can solve the function from the row containing the cell.

The third step utilizes the information from the functional

[2]Also published on the web at http://www.me.utexas.edu/~adl/cfg_grammar.htm.

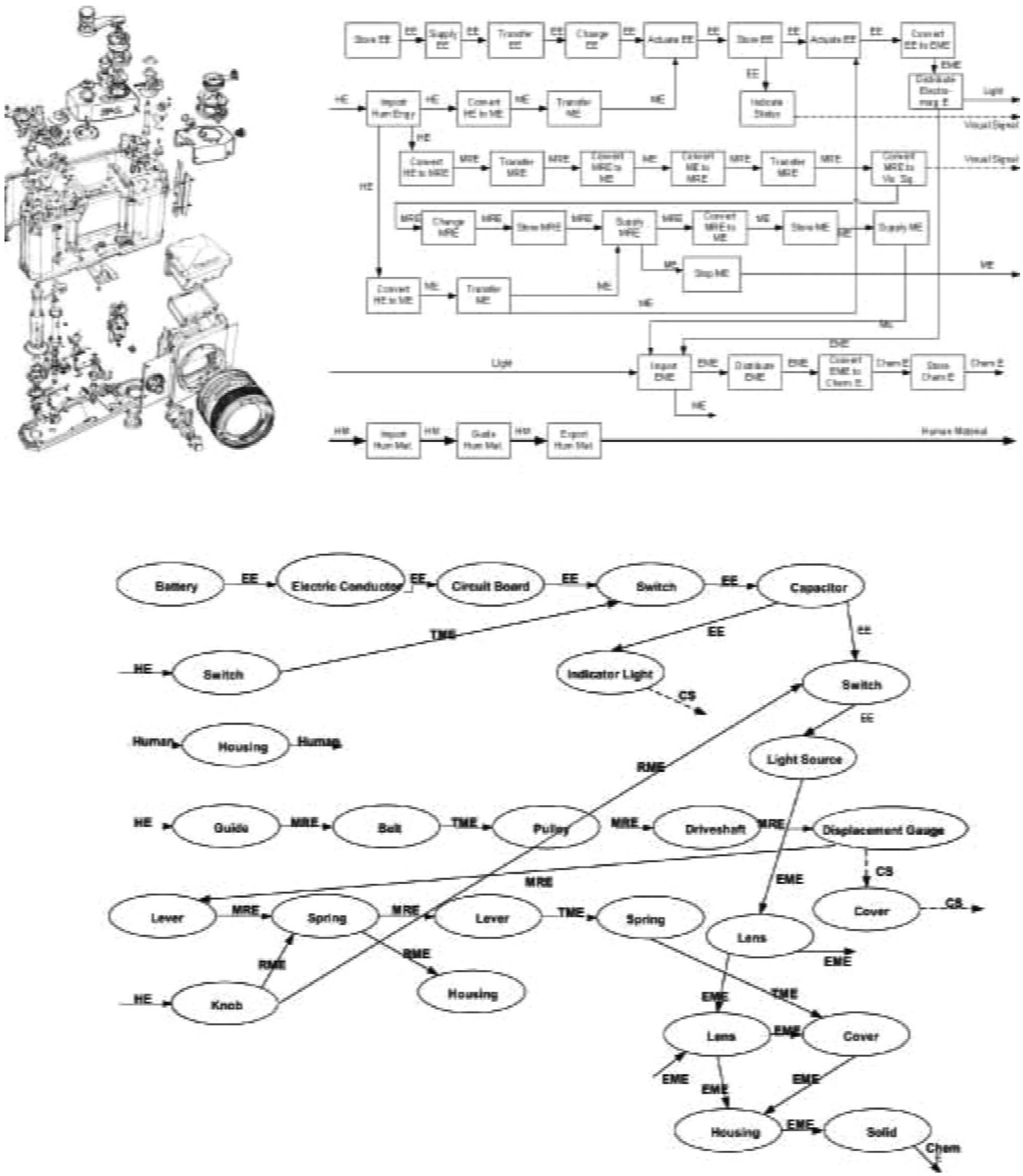

Fig. 3 The function structure, the exploded view, and the configuration flow graph of a camera

model and the FCM to create an unfiltered set of design solutions. Matrix multiplication is used to determine all component pairings that will solve each pairing of connected functions in the flow chain. Tracing every possible "path" of paired connections will give a list of all theoretically possible component chain variations that solve the function chain initially input.

The next step uses additional design knowledge gathered from the design repository to define the compatibility between components in the examined products. As each product is reverse engineered, information regarding the connection between components is extracted from and stored for reuse. Component-component compatibility matrices for individual or specified groups of products can easily be generated from the repository information. Nonzero cell entries in the component-component matrix (frequently called a DSM) indicate that the component from the column containing the cell has been directly connected to the component from the row containing the cell in an existing product.

Finally, the component compatibility information contained in the DSM is used to "prune" the trees of all possible component connections previously computed. Component connections in each solution chain are checked for known compatibility using the

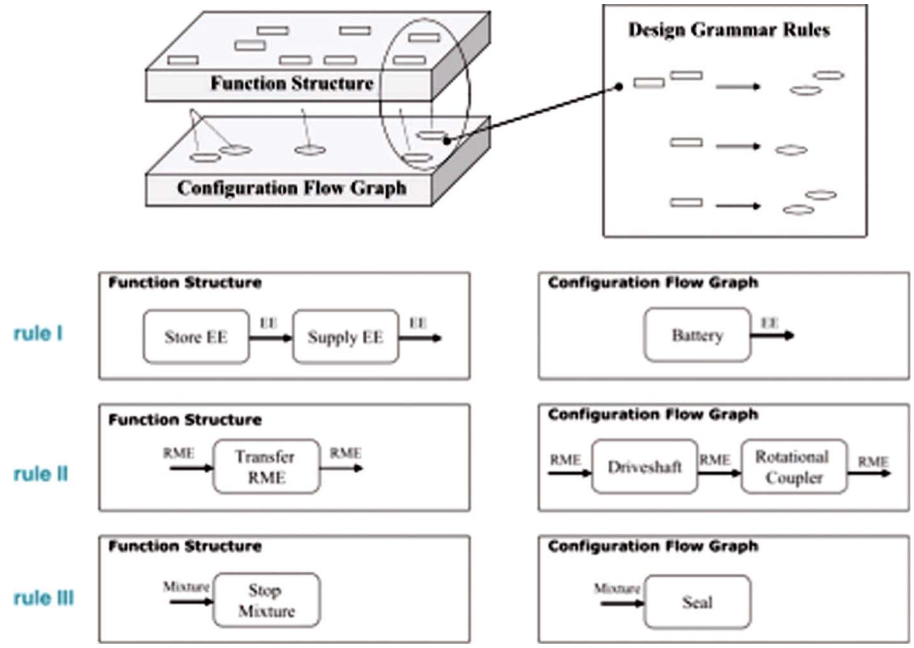

Fig. 4 Illustration of rule derivation from empirical analysis of consumer products

stored connection information from the DSM. This technique uses the "experience" contained in the repository to filter out potentially inadequate concept variants and reduce the set of possible concept variants down to a more manageable size. After the matrices are filtered, we can once again trace every path of possible components to generate a list of feasible component chains that solve the input function chain.

The described algorithm is a method that quickly produces a set of conceptual designs for a new or redesigned product. Functions comprising a proposed product's functional model are mapped to lists of components that are capable of solving each function. The tree of possible component chains is then pruned by eliminating infeasible component connections according to historical component compatibility. This filtered set of component chains is then presented to the designer for further analysis.

5 Discussion

In both conceptual design applications, the proposed component taxonomy plays an essential role. The definition of function based hierarchical component classes provides a consistent means of representing component design knowledge that is then used for computational design synthesis. Specifically, the taxonomy allows us to capture specific relationships between functional needs and components that are used to fulfill them. This facilitates the creation of computational tools with the following characteristics.

Table 1 Summarized results for generating a bottle capping machine using the graph grammar

Example problem	Bottle capping machine
Design problem	
No. of subfunctions in input	28
No. of flows in input	34
Design generation details	
No. of different grammar rubes invoked	42
Total No. of grammar rules invoked	53
No. of candidates generated	339,168
No. of different components used in synthesis	28
List of components	
conductor, switch, wire, cord, electric motor, knob, handle, belt, gear, shaft, agitator, rotational coupler, sprocket, support, lever, link, damp, worm gear, pulley, spring, housing, key, punch, container, sled, guide, piston, hinge	

(1) *Successfully follow the function-based design synthesis process*. As illustrated by the applications, the component terms serve as fundamental building blocks for automatically searching solution principles given functional specifications. In the first application, the taxonomy provides a language to consistently represent component design knowledge and subsequent development of the configuration flow graph representation that captures topological component connectivity information in a computable, graph-based format. Through the use of this graph-based representation, the taxonomy enables the construction of a knowledge base of designer decisions in the form of grammar rules that can later be used for automatically creating conceptual design configurations. Without the developed taxonomy, it would not be feasible to represent component design knowledge in a consistent manner, preventing the development of the grammar and the computational tool that resulted. In the second application, the method of concept generation relies on manipulating matrices and applying combinatory principles to the resulting matrix solutions; therefore, a concise set of component classifications is essential to minimizing the computation time required to generate results. Without a concise taxonomy to classify and collapse component terms in the FCMs and DSMs, the size of each matrix generated by the repository rapidly becomes too large to effectively utilize and the output becomes encumbered by the number of component combinations. Additionally, without some way to consistently define components in a manner that helps ensure exclusivity in term definitions, the quality of results returned degrades when the components that comprise each generated conceptual solution are indistinct and ambiguously defined.

(2) *Allow designers to represent design problems at various degrees of abstraction*. This enables a designer to switch between the function (problem) and component configuration (solution) domains at multiple levels. This knowledge is critical for the concept development stage of product

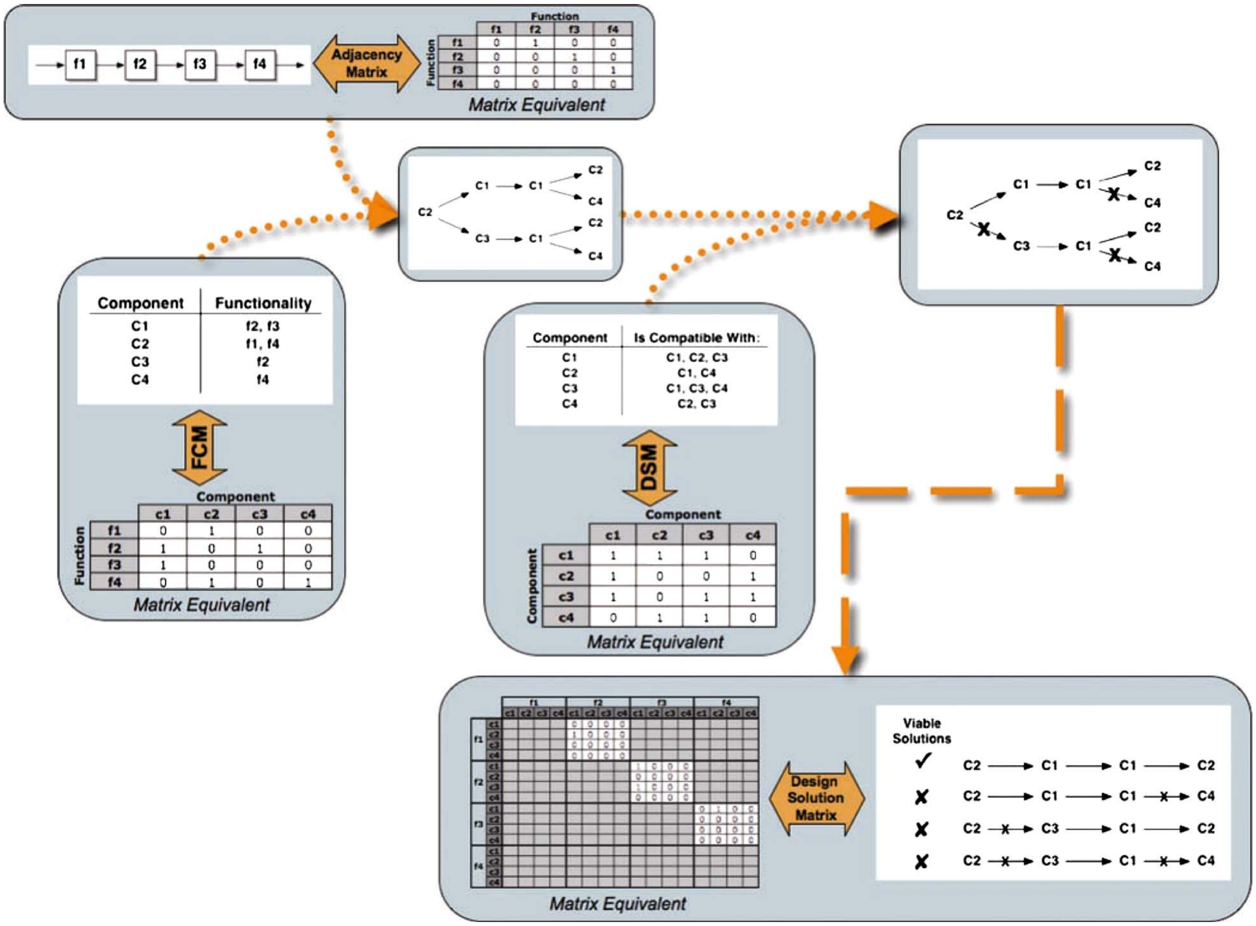

Fig. 5 Brief visual summary of the matrix-based concept generation approach

design, where a designer needs to consider as many functionally similar yet conceptually unique solutions as possible and also requires a certain degree of flexibility in representing the problem at various degrees of abstractions. For example, a designer may think of a "separator" in response to a functional requirement, and can later decide to choose between a "blade" and an "abrasive" (two component concepts structured in the same branch of the hierarchy) as the problem definition and the design requirements mature.

(3) *Are computationally efficient*. Following the guidelines of "completeness" and "exclusivity" eliminates information redundancy and provides an effective encoding scheme for representing component design knowledge. This prevents an undesired combinatorial expansion during automated solution generation. By classifying the 3875 component artifacts using the proposed taxonomy of 135 distinct abstract terms, the size of the FCM and DSM generated from the web-based repository are significantly reduced in size (97.21% reduction in the FCM and 99.92% reduction in the DSM).

Without a rigorously defined component taxonomy, the usefulness of the two automated design tools would be significantly reduced. These applications substantiate how the component taxonomy can be utilized to enhance our abilities at computationally supporting the concept generation phase of the design process.

6 Conclusions

This paper describes a hierarchical framework that is constructed to help guide the classification of components and extend previously presented work toward a component naming convention that led to a flat list of 135 distinct generic component terms [31]. In addition, the framework presented here uses primary and secondary levels of specification coupled with a robustly defined procedure to help identify the appropriate placement of terms into the hierarchy while maintaining the goals of completeness and exclusivity in component coverage. Under this proposed framework, components of widely varying levels of complexity (e.g., an electric wire versus an electric motor) may both be placed within the hierarchical structure, as long as the black box functionality may be limited to a single function contained within the functional basis list of terms. Additionally, components that exhibit functionality directly vital to the functioning of a product (e.g., a plug and cord) are not distinguished from components that only exhibit functionality that supports the function of a product in a more indirect manner (e.g., a bracket that secures an electric motor in place). Finally, while component definitions sometimes reference a component's form or its method of fabrication, such information is not used within the component hierarchy. For the components classified thus far, complexity, type of functionality (i.e., whether it directly or indirectly works to solve conceptual functionality), and other such characteristics do not seem to negatively impact the proposed framework, and the selected hierarchy depth of four (e.g., connectors, couplers, fasteners, and screw) is deemed sufficient for the kind of configuration-based conceptual design problems targeted by the application tools. However, the proposed framework could be easily adjusted to fit into a larger hierarchical framework where other component characteristics that are considered appropriate may be added as subgroups to the proposed hierarchy. For example, if one wishes to represent form or material

Table 2 Summary table showing the number of design solutions found in both the student generated morphological matrices and the morphological matrices derived from the concept generator results

		# of Solutions Generated	# of Solutions that Appear in Both Concept Generation Methods	% Overlap with Alternatively Produced Solutions
Power Supply Design Scenario	Student Generated Morphological Matrix with Solutions Classified Using Component Name Classifications	43	19	44.19%
	Automatically Generated Solutions Using Concept Generator Software	150	19	12.67%
Thermal Mug Design Scenario	Student Generated Morphological Matrix with Solutions Classified Using Component Name Classifications	79	35	44.30%
	Automatically Generated Solutions Using Concept Generator Software	152	35	23.03%
Wall Climber Toy Design Scenario	Student Generated Morphological Matrix with Solutions Classified Using Component Name Classifications	25	12	48.00%
	Automatically Generated Solutions Using Concept Generator Software	109	12	11.01%

specific attributes of the screw class, a fifth level can be introduced to the hierarchy, which may include screw classes such as machine screw, set screw, and wood screw.

In addition to establishing a method of consistently achieving complete and exclusive coverage of the component space, the hierarchical taxonomy also establishes a means to distinguish traditionally similarly named components that, in fact, have very different functionality. A formal naming structure could be used to distinguish common component names that may be misleadingly similar (e.g., a wheel used as a control device to, for example, steer a car versus a wheel that is fixed to an axle and allows for an object, such as a vehicle, to roll along the ground). Since the primary motivation behind the creation of the component taxonomy is to assist designers during the early phases of design, a hierarchy organized by functional purpose incorporates a level of abstraction that will allow functionally similar but distinct components to be considered for a design. By following the presented procedure and utilizing the proposed hierarchical structure where components are grouped together by functional purpose and distinguished by functional embodiment, it is postulated that the goals of completeness and exclusivity of term coverage will also be effectively maintained. To maintain the completeness criterion, we closely monitored the derivation rate for newly defined component terms. After studying over a hundred products, we observed a reduction in the number of newly defined component terms as greater numbers of products were cataloged into the repository. It is likely that the number of component terms will slowly converge to a finite value through the dissection of new products.

For validating the exclusivity criterion, we have conducted experimental studies [30,32,33] in order to assess the effectiveness of the proposed taxonomy and the automated design tools that utilize it. In these studies, we compare the design solutions generated by the automated tools using the language afforded by the taxonomy with the design solutions created by human designers following the traditional concept generation methods. Initial studies, as supported by the two computational applications, has shown that the 135 component terms provide a consistent means for representing and organizing component design knowledge and that the automated design tools generate solutions comparable to human conceptual designing [34–36]. As an example, Table 2 shows a summary of the number of overlapping design solutions seen for three different design scenarios.

Morphological matrices generated by student designers are compared with morphological matrices produced by the second concept generator application. For instance, if we look at the human-powered power supply design scenario data, of the 43 distinct solutions produced by the students, 19 matching solutions were common to the automated concept generator, meaning 44.19% of the student generated solutions were contained in the automatically generated solution set. Of the 24 remaining solutions produced by the students, six were not definable under the current version of the component naming scheme. Inspection of the results returned by the automated concept generator that did not overlap with the results generated by the students showed an overwhelming majority of viable alternatives. Only a few instances of obvious incorrect matches were identified, and each was linked back to data entry mistakes that occurred while the repository was being populated with product information.

The results of developing such a thorough taxonomy are not solely useful to computational design synthesis research. Arriving at a standard set of names to represent electromechanical components could be useful to vendors selling OEM components. Furthermore, a wealth of dynamic simulation tools such as DYMOLA [34], WORKING MODEL [35], and SCHEMEBUILDER [36] could use the fundamental component concepts to allow for easy user familiarization and interoperability. Finally, the interactions between engineers both in industry and academia could be ameliorated by an established set of standard names.

7 Future Work

To build on the work presented here, future work will include establishing port templates that may be used to help build up more complete conceptual ideas during the early stages of conceptual design. By knowing the number and types of ports, a component term typically has, software may be used to help guide the evolution of a full conceptual idea, including parts needed to indirectly support the functionality of other components. Additionally, design measure estimates (such as measures of potential failures, manufacturability, cost, size, and performance) could be determined across each component group and used to help guide concept selection early in the design process. Additionally, we aim to improve the tree based hierarchy by defining relations between component concepts that will allow inheriting properties from two exclusive (same or higher level) concepts.

Other future work includes creating a forum for the discussion of new and existing component terms, their placement within the hierarchical taxonomy, and even the organization of the hierarchical taxonomy. What we have developed over the past several years is a well-defined proposal for the representation of electromechanical component knowledge in the context of conceptual design. Of course, to arrive at the standard itself, the proposed taxonomy needs to be evaluated by a larger community. Fortunately, advances in the internet and object-oriented programming offer some respite in achieving such a goal. Accordingly, we plan on publishing the taxonomy on the UMR design repository website and initiating an open forum discussion to define and classify components as a worldwide community effort.

Finally, the work presented here is focused mainly on components found in consumer products. Additional work should look at other design domains and identify how the hierarchy should be altered or expanded to include a broader range of component types. However, as the number of component terms grows, ensuring the uniqueness of every taxonomic concept becomes challenging. Here, automated tools can assist in controlling data entry to the taxonomy and help maintain the exclusivity criterion and the overall consistency of the taxonomy.

Acknowledgment

The authors would like to thank the National Science Foundation for supporting this work under Grant No. IIS-0307665.

References

[1] Strawbridge, Z., McAdams, D., and Stone, R., 2002, "A Computational Approach to Conceptual Design," *Proceedings of the DETC2002/DTM-34001*, Montreal, Canada.
[2] Bryant, C., Stone, R., McAdams, D., Kurtoglu, T., and Campbell, M., 2005, "A Computational Technique for Concept Generation," *Proceedings of the ASME 2005 International Design Engineering Technical Conference*, Long Beach, CA.
[3] Kurtoglu, T., Campbell, M., Gonzales, J., Bryant, C., Stone, R., and McAdams, D., 2005, "Capturing Empirically Derived Design Knowledge for Creating Conceptual Design Configurations," ASME 2005 International Design Engineering Technical Conference, Long Beach, CA.
[4] Chakrabarti, A., and Bligh, T., 1996, "An Approach to Functional Synthesis of Mechanical Design Concepts: Theory, Applications and Emerging Research Issues," Artif. Intell. Eng. Des. Anal. Manuf., **10**, pp. 313–331.
[5] Campbell, M. I., Cagan, J., and Kotovsky, K., 1999, "A-Design: An Agent-Based Approach to Conceptual Design in a Dynamic Environment," Res. Eng. Des., **11**(3), pp. 172–192.
[6] Hubka, V., and Ernst Eder, W., 1984, *Theory of Technical Systems*, Springer-Verlag, Berlin.
[7] Pahl, G., and Beitz, W., 1988, *Engineering Design: A Systematic Approach*, Springer-Verlag, Berlin.
[8] Suh, N., 1990, *The Principles of Design*, Oxford University Press, New York.
[9] Ullman, D., 1997, *The Mechanical Design Process*, 2nd ed., McGraw-Hill, New York.
[10] Ulrich, K., and Eppinger, S., 1995, *Product Design and Development*, McGraw-Hill, New York.
[11] Otto, K., and Wood, K., 2001, *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice-Hall, Englewood Cliffs, NJ.
[12] Liang, V., and Paredis, C. J. J, 2004, "A Port Ontology for Conceptual Design of Systems," ASME J. Comput. Inf. Sci. Eng., **4**, pp. 206–217.
[13] Wallace, A. P., 1995, *The Modelling of Engineering Assemblies Based on Standard Catalogue Components*, University of Bath, Bath, UK.
[14] Culley, S. J., and Webber, S. J., 1992, "Implementation Requirements for Electronic Standard Component Catalogues," Proc. Inst. Mech. Eng., Part B, **206**, pp. 253–260.
[15] Hicks, B. J., Culley, S. J., and Mullineux, G., 2005, "The Modeling of Engineering Systems for Their Computer Based Embodiment With Standard Components," ASME J. Mech. Des., **127**, pp. 424–432.
[16] Uschold, M., and Gruninger, M., 1996, "Ontologies: Principles, Methods, and Applications," Knowl. Eng. Rev., **11**, pp. 93–136.
[17] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R., 1991, "Enabling Technology for Knowledge Sharing," AI Mag., **12**(3), pp. 36–56.
[18] Horváth, I., Dorozsmai, K., and Thernesz, V., 1994, "A Feature-Object-Based Practical Methodology for Integration of Conceptual and Morphological Design," *Lancaster International Workshop on Engineering Design CACD '94*, Lancaster University, pp. 131–149.
[19] Horváth, I., Vergeest, J. S. M., and Kuczogi, G., 1998, "Development and Application of Design Concept Ontologies for Contextual Conceptualization," ASME Paper No. DETC98/CIE-5701.
[20] Kitamura, Y., and Mizoguchi, R., 2003, "Ontology-Based Description of Functional Design Knowledge and Its Use in a Functional Way Server," Expert Sys. Applic., **24**, pp. 153–166.
[21] Li, Z., Anderson, D., and Ramani, K., 2005 "Ontology-Based Design Knowledge Modeling for Product Retrieval," *Proceedings of the 2005 International Conference on Engineering Design, ICED 2005*, Melbourne, Australia.
[22] Stahovich, T. F., Davis, R., and Shrobe, H., 1993, "An Ontology of Mechanical Devices," *Proceedings of the 1993 AAAI National Conference on Artificial Intelligence*.
[23] Szykman, S., Racz, J., and Sriram, R., 1999, "The Representation of Function in Computer-Based Design," *Proceedings of the 1999 ASME Design Theory and Methodology Conference*, Las Vegas, NV.
[24] Fenves, S. J., 2001, "A Core Product Model For Representing Design Information," National Institute of Standards and Technology, Report No. NISTIR6736.
[25] Terpenny, J., 1997, "An Integrated System for Functional Modeling and Configuration in Conceptual Design," *Proceedings of the Seventh International Flexible Automation and Intelligent Manufacturing Conference*, Middlesbrough, UK, Jun. 25–27, pp. 69–80.
[26] Sikand, A., and Terpenny, J., 2004, "A Web-Based Environment for Managing Product Knowledge," International Symposium on Collaborative Technologies and Systems (CTSO 4), San Diego, CA.
[27] Hirtz, J., Stone, R., McAdams, D., Szykman, S., and Wood, K., 2002, "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts," Res. Eng. Des., **13**(2), pp. 65–82.
[28] Chenhall, R. G., 1978, *Nomenclature for Museum Cataloging: A System for Classifying Man-Made Objects*, American Association for State and Local History, Nashville, TN.
[29] The University of Missouri at Rolla, Design Knowledge Repository, http://function.basiceng.umr.edu/repository.
[30] Kurtoglu, T., 2007, "A Computational Approach to Innovative Conceptual Design," Ph.D. thesis, The University of Texas at Austin, Austin.
[31] Greer, J., Stock, M. E., Stone, R., and Wood, K., 2003, "Enumerating the Component Space: First Steps Towards a Design Naming Convention for Mechanical Parts," *Proceedings of the DETC2003*, Chicago, IL.
[32] Bryant, C., McAdams, D., Stone, R., Kurtoglu, T., and Campbell, M., 2006, "A Validation Study of an Automated Concept Generator Design Tool," ASME 2006 Design Theory and Methodology Conference, Philadelphia, PA.
[33] Bryant, C. R., Pieper, E., Walther, B., Kurtoglu, T., Stone, R. B., McAdams, D. A., and Campbell, M. I., 2006, "Software Evaluation of an Automated Concept Generator Design Tool," 2006 American Society for Engineering Education Annual Conference & Exhibition, Chicago, IL, Paper No. 2005-1758.
[34] Dynasim Software, 2004, "Dymola—Dynamic Modeling Laboratory—Multi-engineering Modeling and Simulation," http://www.dynasim.com/www/DymolaBrochure.pdf.
[35] MSC Software, 2004, "Working Model—The World's Most Popular 2D Motion Software," http://www.krev.com/overview.html.
[36] Bracewell, R. H., and Sharpe, J. E. E., 1996, "Functional Descriptions Used in Computer Support for Qualitative Scheme Generation—Schemebuilder," Artif. Intell. Eng. Des. Anal. Manuf., **10**(4), pp. 333–346.