Presented at the First International Workshop on Formal Methods in Design, Colorado Springs, Jan. 14-17, 1990.

TOWARD A VOCABULARY FOR CLASSIFYING RESEARCH IN MECHANICAL DESIGN AUTOMATION

G.L. Snavely, L.P. Pomrehn, P.Y. Papalambros

Department of Mechanical Engineering and Applied Mechanics
The University of Michigan
Ann Arbor, Michigan 48109, USA

ABSTRACT

Research efforts for automating the design process of mechanical artifacts have been intensified in recent years. A variety of approaches have been proposed. However, as is usual in a new field, no unified theory or terminology has yet emerged. This situation is complicated further by lack of general agreement on what constitutes "design". Evidently, some common framework or vocabulary is necessary for researchers to be able to communicate and compare their efforts. This paper attempts to outline such a vocabulary and examines, by example, how it could be used to review current published research in mechanical design automation.

1 INTRODUCTION

When reviewing research papers in the field of mechanical design automation, a variety of knowledge representation methodologies and problem solving strategies is found. Each methodology and strategy introduces distinct design terminology. The attendant wide variety of such terminologies makes it difficult to compare and contrast work on similar topics. After all, it is not immediately obvious what a "hierarchy of specialists" [5] and "agents and critics" [31] have in common. Even more difficult is the task of distinguishing areas in which substantial research remains to be done from areas that are heavily populated by researchers.

Due to the extent of design terminologies that have been suggested and used, it is highly desirable to develop some sort of *design vocabulary* into which all (or almost all) others can be readily translated. The goal of such a vocabulary is not to standardize knowledge representation methodologies or problem solving strategies, but to simplify comparisons of existing methodologies and strategies, and to provide a common way of talking about them. In the present paper, a design vocabulary for classifying mechanical design is proposed, and then applied specifically to research in mechanical design automation, by way of reviewing several published research articles.

Other methods for classifying work in mechanical design have been proposed. Dixon et. al. [10] classify mechanical design research by *design problem*, using a methodology based on initial and desired states of knowledge. Ullman [34] builds on this and other work [18] [28] by proposing a broader classification scheme. His taxonomy not only classifies work in mechanical design according to design problem, but also according to *research method, environment, and design process*. The design vocabulary described in this paper presents an alternate means of classifying mechanical design by *design process*. This vocabulary has evolved from a mechanical design model that views design as *an iterative process of combining successively less abstract structural and functional entities into useful combinations*. Although there is no universal agreement on this design model, or a definition of design derived from it, the model is useful in practice and appears natural to most design engineers.

The design vocabulary provides a means for discussing two independent classifications of a design process: *design type* and *design activity.* In the subsequent sections, following the description of the design vocabulary, these two classifications are incorporated into a two-dimensional matrix, called the *classification framework*. Research taking place in mechanical design automation is then placed into this framework, with each item of research discussed using the design vocabulary. The result of placing research into the framework illustrates how the design vocabulary helps in locating both densely and sparsely populated areas of mechanical design research.

2 DESIGN VOCABULARY

The design vocabulary provides an environment for distinguishing a design task by design type and design activity. *Design type* classifies a design task according to its degree of inventiveness, while *design activity* is concerned with the degree of structural representation employed. "Structure" here is related to abstraction, according to our adopted model of the design process: the less abstract a design description is, the more structure it has. The vocabulary supporting these two design classification methods is discussed in the next two sections.

2.1 Design Type

Classifying research by design type is a means of grouping together work taking place at similar levels of design inventiveness. In order to distinguish between different design types, the terms in the design vocabulary used to describe design types must be clearly defined. These terms include *catalog, catalog entry, fixed topology, variable topology,* and *slot.*

A *catalog* is simply a database to which the design system has access. The catalog contains individual pieces of data called *catalog entries*. Catalog entries may include the specific artifacts found in traditional engineering catalogs, as well as more abstract objects. For instance, a spring catalog might contain, in order of increasing abstraction, (a) spring #SPR1-1 from the Berg catalog [4], (b) an ends-squared, stainless steel, compression spring, (c) a steel compression spring, (d) a spring, and (e) a device that outputs an opposing force proportional to its deflection. Additionally, a catalog entry may be a generic concept, such as a dimension or material. The utility of including such generic concepts is demonstrated below. It is worth noting that a catalog, as defined here, has no inherent structure, although all of the research reviewed in Section 3 imparts some structure to the catalog.

A *topology* is a structure representing the relationship between slots. A *fixed topology* is an invariant structure of slots, while a *variable topology* places no restriction on the structuring of slots. A *slot* is a location into which a catalog entry can be placed. Furthermore, slots may have the ability to filter out inappropriate catalog entries, so that the term *slot filters* implies slots with such an ability. An example of a fixed topology problem is gear pair selection. The topology contains two slots, each to be filled with an appropriate gear. A variable topology problem would be gear train design. In gear train design, stages added to the gear train are represented as additional slots in the topology. In both the gear pair and the gear train example, slot filters insure that adjacent gears mate properly.

Using the above vocabulary, the four mutually exclusive types of design are now defined.

> Type 1 Design, called *invention*, is the process of creating a new entry to a catalog that cannot not be created by combining existing catalog entries.
>
> Type 2 Design, called *innovation*, is the process of filling the slots of a variable topology with catalog entries.
>
> Type 3 Design, called *routine design*, is the process of filling the slots of a fixed topology (or a predetermined set of fixed topologies) with catalog entries.
>
> Type 4 Design, called *procedural design*, is an algorithmic process for filling the slots of a topology with a single combination of catalog entries.

While these definitions are far from complete, they do provide the groundwork for a general discussion of design automation. In order to illustrate best the four types of design, each type is now illustrated in the context of a common example.

The example concerns the design of an automobile windshield wiper system. One possible topology for such a system is a sequential chain of slots containing *drive*, *transmission*, *arm*, and *blade* filters (Figure 1). The drive is the motion source of the wiping system. The transmission converts one type of motion to another, as in rotary to linear, or unidirectional to bidirectional conversion. The arm is a structural component connecting the transmission to the blade. The blade wipes water from the windshield. In order to keep the example manageable, the number of catalog entries is greatly limited. Catalog entries are placed into groups corresponding to slot filters through which they are allowed to pass. The entries in the catalog corresponding to the drive slot filter are a continuous motion actuator and a reciprocating motion actuator. Under transmission, entries include pulleys, gears, and four-bar linkages. For the arm, the two entries are a plastic arm and a steel arm. Finally, under blade, the entries are a long blade and a short blade. The catalog for this example therefore contains 9 entries (Figure 2).

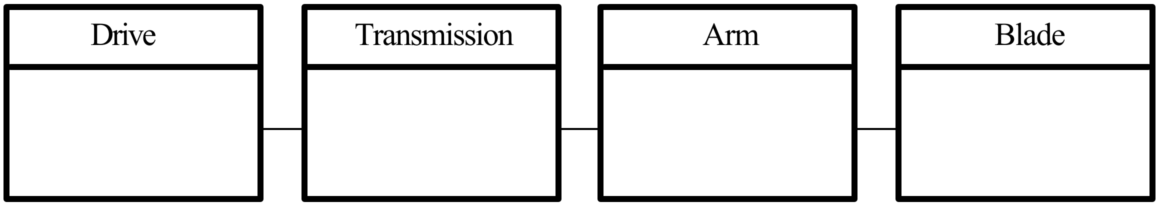

Figure 1: Automobile Wiper System Topology

Continuous Motion Actuator
Reciprocating Motion Actuator

Pulleys
Gears
Four-Bar Linkages

Plastic Arm
Steel Arm

Long Blade
Short Blade

Figure 2: Automobile Wiper System Catalog

A possible Type 1 Design for the windshield wiper system incorporates a cam and follower transmission (Figure 3). A cam and follower transmission does not exist in the catalog, and no combination of existing catalog entries would generate it, thus a cam and follower must be added to the catalog.

Continuous Motion Actuator
Reciprocating Motion Actuator

Pulleys
Gears
Four-Bar Linkages
Cam and Follower

Plastic Arm
Steel Arm

Long Blade
Short Blade

Figure 3: Type 1 Design: Expanded Catalog

For Type 2 Design, the topology is allowed to vary. One way to vary the topology is to eliminate a slot, say the transmission. Thus, one potential Type 2 Design topology contains slots for drive, arm, and blade catalog entries. In contrast to the elimination of a slot, another topology variation is duplicating a slot a number of times. For instance, increasing the number of blades from one to two, leads to a Type 2 Design for a dual blade wiper system (Figure 4).

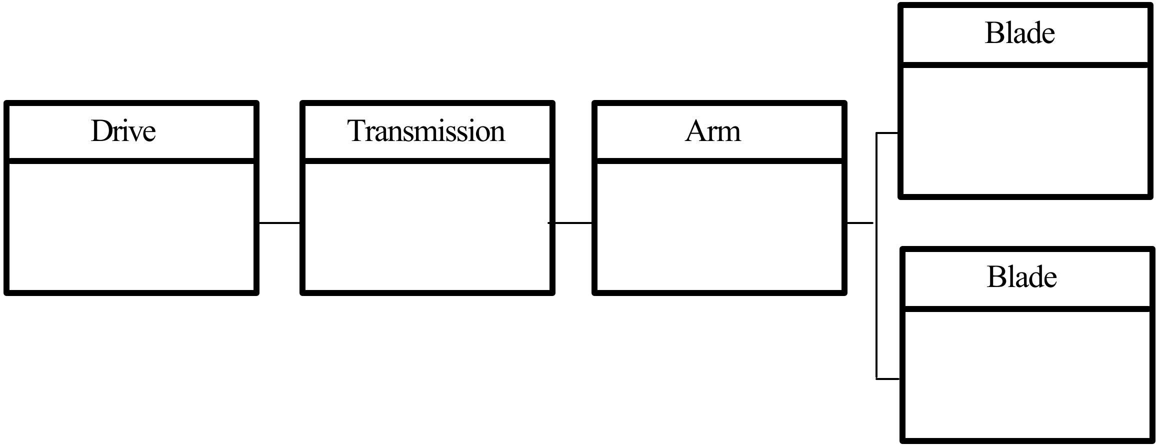

Figure 4: Type 2 Design: Slot Duplication

In Type 3 Design, the topology is fixed. That is, the drive slot must be connected to the transmission, the transmission to the arm, and the arm to the blade. In addition, no slots can be duplicated or deleted. Thus, any of the 24 (2x3x2x2) allowable combinations of catalog entries is a feasible solution to the design problem (Figure 5). The combinatorial nature of both type 2 and type 3 design points to the need for intelligent ways of decreasing the search space and systematically identifying good solutions.

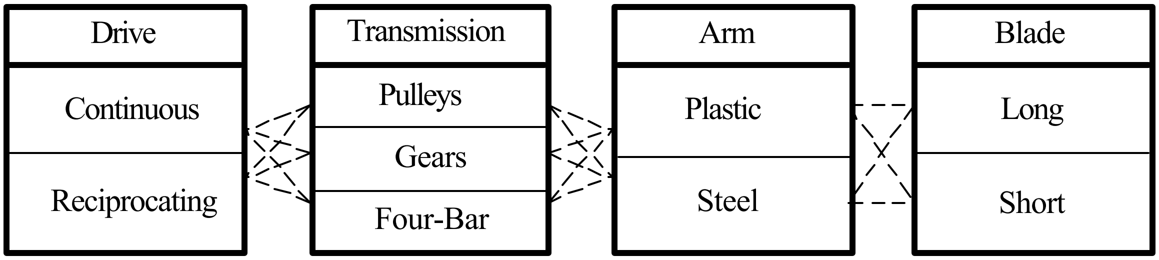

Figure 5: Type 3 Design: Allowable Slot Combinations

For Type 4 Design, a procedure for choosing each catalog entry is completely specified. Ruling out iterative or combinatorial approaches, such as search or generate and test, causes Type 4 procedures to move directly toward a single combination of catalog entries (Figure 6). An example might progress as follows. Due to the size of the windshield, a long blade is required. The choice of a long blade requires selecting both a steel arm (for strength) and a gear transmission (for appropriate gear ratio). Finally, the selection of a gear transmission, along with the implicit requirement of a reciprocating wiper system, requires the choice of a reciprocating motion actuator.

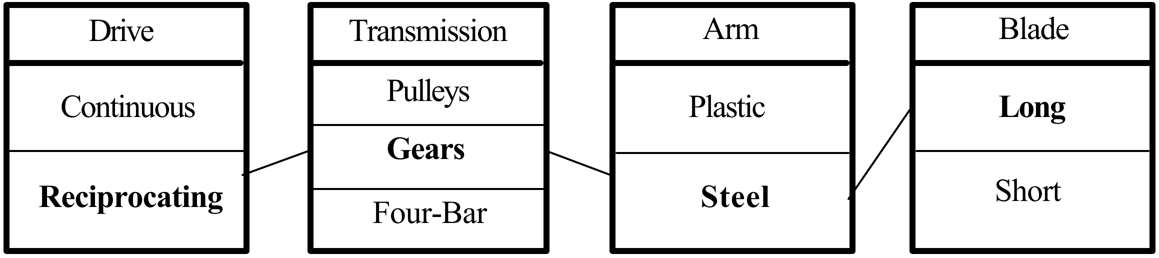

Figure 6: Type 4 Design: Filled Wiper System Topology

2.2 Design Activity

A second means of classifying a design task examines the degree of structure imparted to the design under consideration and is called *design activity.* While the design type deals with the inventive content of a particular approach to a design problem, design activity deals with the level of abstraction at which the design task is addressed. Three distinct levels of abstraction are identified. These levels are *function, transition,* and *form.* As with the four design types, these three design activities are mutually exclusive, at least to the extent that one can distinguish function from form. However, this exclusive nature in no way insures a unique classification of each researcher's contribution. Rather, a majority of the work reviewed operates within two or three design activities. Each design activity is now described in greater detail and illustrated through the use of the wiper system example.

The *function* activity is concerned with design systems that convert between different *functional descriptions* of the artifact. Functional descriptions are descriptions of what an artifact should do or how it should behave, and are not concerned with how the functionality is realized. In order to demonstrate how a design system within the function activity operates the wiper system example is broadened. For instance, an input functional description is "clear all fluid and solid particles from the windshield of an automobile." An automated design system, working within the function activity, might replace this functional description with the more specific description "produce a motion to drive a debris-clearing element which removes both fluid and solid particles from the windshield of an automobile." Further refinement could replace the motion-producing element with the previously mentioned drive and transmission elements.

The *transition* activity deals with two distinct, yet similar, transformations: function-to-form and form-to-function. A function-to-form transformation converts a functional description into one or more forms satisfying the functionality. These forms describe attributes such as the shape, structure, or size of the artifact. In the wiper example, the drive element could be implemented in the form of a reciprocating motion actuator and the debris-clearing element implemented in the form of a blade. Although many view design as a transformation from function to form, the reverse form-to-function transformation is also important in mechanical design. Observing the form of an artifact often reveals much about its functionality. This functionality provides feedback to a standard function to form design process.

The *form* activity includes systems which alter the form of an artifact. Such a system may define the artifact's configuration by replacing two catalog entries with a single entry. For example, a design system could convert a continuous motion actuator and a four-bar linkage transmission into a reciprocating motion actuator. Another system may alter the form of the artifact by changing its dimensions. An example is a system that assigns a length to the wiper blade, trading off area covered with power required.

2.3 Classification Framework

As mentioned in the introduction, the design vocabulary proposed here has evolved from a specific model of mechanical design. This model views design as an iterative process of combining successively less abstract structural and functional catalog entries into useful combinations. While the vocabulary results directly from that model, in no way does it interfere with other models of the design process. Rather, the design vocabulary provides a uniform way of considering such models. Researchers who view design as a transformation (function to form, abstract to specific, etc.), as a combinatorial process (morphological analysis), or as a search technique, should have little difficulty expressing their ideas in the proposed vocabulary.

The best way to show the versatility of the design vocabulary is to use it to describe some of the research reported in mechanical design automation. In order to proceed with a review in a structured manner, a "classification framework" is introduced. The *classification framework* is a two-dimensional matrix with axes defined by design type and design activity. The classification framework may serve to illustrate the density of research efforts across all design types and activities. A number of research papers are reviewed using the design vocabulary. Each paper is positioned in the appropriate location within the classification framework. As the review is intended to illustrate the utility of the design vocabulary, it is neither exhaustive nor complete; in particular, no attempt is made to express relative importance of such research.

3 RESEARCH REVIEW USING THE DESIGN VOCABULARY

The papers reviewed below represent a cross section of work published in the field of mechanical design automation. The reviews are grouped into four sections according to design type, the primary axis of the classification framework. Within each type, reviews are further classified by design activity, the secondary axis of the classification framework. Note that not every activity is represented within each design type. Figure 7, the classification framework, illustrates the placement of reviewed papers. An additional section, discussing "design support" tools, follows the four sections on design types.

Inevitable disagreements about the location of a research paper in the classification framework are to be anticipated. Two potential sources of misrepresentation lend validity to any such disagreement. First, just like translation problems between spoken languages, not all design terminologies can be translated into a given design vocabulary without some loss of content. Second, the review of a paper does not always give complete insight into the content of the research. Responsibility of any such misrepresentation is of course assumed by the present authors. Another disagreement may come from omission of possibly related work. One aspect of omission results from the present focus on the mechanical design domain. While research in the fields of architecture, circuit, and software design fit within the framework, the scope of this paper is limited to the authors' field of expertise. A second aspect of omission results from the increasingly large amount of research in the field and our attempt to use the review only as an illustration of the proposed design vocabulary.

	TYPE 1 INVENTION	TYPE 2 INNOVATION	TYPE 3 ROUTINE	TYPE 4 PROCEDURAL
FUNCTION		Freeman and Newell [15] Ulrich and Seering [37]		
TRANSITION		Cagan and Agogino [6] Duffey and Dixon [12] Doyle [11] Joskowicz and Addanki [21]	Brown and Chandrasekaran [5] Hoeltzel and Chieng [17] Kota et. al. [23] Mittal et. al. [26] Murthy and Addanki [27] Ulrich and Seering [35,36]	Papalambros and Wilde [29]
FORM		Dyer and Flowers [13,14]	Arciszewski and Aktan [2] Rehg et. al. [31] Ward and Seering [38,39]	

Figure 7: Classification Framework

3.1 Invention

While no research in inventive mechanical design is discussed in the present article, it should not be concluded that none exists. Furthermore, asserting that an automated mechanical design invention system can exist is equally acceptable to the assertion that design will remain ultimately a human activity (perhaps at an appropriate level of abstraction). Without a better understanding of automated innovation, progress toward automated invention will be necessarily slow. Some of the most promising ideas in the area of automated invention appear in the work of Lenat [24], although the concepts are not directly applied to mechanical design.

The definition of invention given earlier appears rather restrictive: invention requires the creation of catalog entries that are not combinations of existing entries. However, this restrictive definition is required in order to prevent combinatorial systems from generating potential "inventions" in an exponential manner. After all, invention must be regarded as more than the exhaustive combination of various forms and functions. Thus, while the definition presented here is a necessary condition, it may not be sufficient to guarantee invention, and the definition itself remains a tentative one.

3.2 Innovation

Innovation traditionally describes designs that are certainly creative, but which are not creative enough to be considered inventions. The definition of this term given in the previous section allows its use in a more rigorous manner. Innovation involves the generation of a potentially infinite set of design topologies. This infinite set consists of topologies that are either finite or infinite in length. Most innovative design systems place some limits, usually in the form of heuristics, on the size of the topologies generated. These limits often resemble rules like "the number of slots in a topology must be less than five" or "no two identical catalog entries may be neighbors in a topology." While heuristics used to control combinatorial explosion may vary widely, if these limits were removed, all innovative design systems could potentially generate an infinite number of topologies.

3.2.1 Function Design Activity

Innovation is the only type of design where a significant amount of research in *functional design activity* is observed. This observation seems intuitively reasonable, as routine and procedural design are each generally associated more closely with form than with function.

Some of the earliest work in the function activity was pioneered by Freeman and Newell [15], who developed a methodology for refining the functional requirements of a design problem. This methodology, called functional reasoning, consists of structures (slots) grouped together by functional connections (topology). Further, Freeman and Newell proposed a series of postulates governing the properties of both the topology and the individual slots.

Building on the concept of functional reasoning, Ulrich and Seering [37] developed examples of schematic representations of functional descriptions. They discussed "schematic synthesis", a process for refining the functional topology of the design problem. Schematic synthesis requires both system input and system output functional constraints, as well as a required relationship between the two, in order to generate a feasible functional topology. Ulrich and Seering applied this methodology to the domain of devices that can be accurately modeled by functional topologies of idealized elements. The elements (catalog entries) in their application include inertias, resistances, capacitances, translators, and gyrators. Applicability to other similar domains appears plausible but has not yet been demonstrated. Schematic synthesis is innovative because the rules for combining functional catalog entries allow the generation of an infinite number of functional topologies.

3.2.2 Transition Design Activity

Most of the research on innovative design involves the conversion of function to form, and is called *transition design activity.*

Duffey and Dixon [12] have developed a "topological redesign module" that combines high-level catalog entries to create a two-dimensional geometrical topology meeting a set of functional requirements. The utility of this methodology is demonstrated in the domain of extruded cross-section geometries. Catalog entries to the geometrical topology consist of elements like walls, webs, and flanges. Duffey and Dixon define heuristics to control the manner in which these entries can be combined, and to create simple topologies before more complex topologies.

Doyle [11] discusses a methodology for hypothesizing the mechanisms of a diverse group of devices including a toaster, a bicycle coaster brake, and a tire pressure gauge. Doyle's catalog contains an equally diverse collection of both functional and structural catalog entries. Functional catalog entries include fluid heat transport, condensation, evaporation, and thermal expansion. Structural catalog entries include rotary couplings, springs, and switches. Given the description of a device's functionality, Doyle's system creates a plausible functional topology. The system then fills the slots in the topology with both functional and structural catalog entries. Subsequent refinement replaces functional entries with structural entries, resulting in a structural topology.

The "First Principle Computational Evaluator" or "1stPRINCE" [6] is a system aimed at designing mechanical structures ("structure" here having the traditional mechanical engineering meaning) from "first principle" equations. The system generates new topologies by splitting an integral in a first principle equation. In a torsional beam example, an integral over the radius of the beam (from 0 to r) is replaced by two integrals covering the same radius (0 to r_1 and r_1 to r). The two regions created constitute a new topology, allowing a composite beam to replace the previously homogeneous beam. The composite beam now requires new catalog entries, in the form of dimensions and material properties, to fill in the empty slots of the expanded topology. While the system converts between forms of the artifact, functional requirements driving the integral splitting procedure makes the system transitional. 1stPRINCE appears limited in the size of the problem that can be attacked. A torsional beam is governed by only a few first principle equations; in general, the integral splitting process leads to combinatorial explosion.

Joskowicz and Addanki [21] use innovative techniques to determine the shape of kinematic pairs. Functional descriptions are converted into form as the shape of the kinematic pair is determined. Functional requirements arise from differences between the desired functionality and the functionality provided by an existing, user input topology. These functional requirements alter the topology of the kinematic pair. New slots in the altered topology are filled with shape elements satisfying the functional requirements. While the system of Joskowicz and Addanki deals solely with kinematic pairs, the potential shapes of such pairs are infinite.

3.2.3 Form Design Activity

The *form design activity* deals with systems whose main operation is to alter the form of an artifact. Note that these changes of form may be driven by functional requirements.

The system EDISON [13,14] performs innovation by applying heuristics to generate new structural topologies. Generalization, analogy, and mutation heuristics are described, although only mutation heuristics appear to have been coded at present. The example used involves design of doors. The mutation heuristics employ alteration and combination to create new door topologies. For example, the number of hinges is altered, with multiple hinges ultimately leading to the idea of a continuous hinge. However, EDISON never really understands the concept of a continuous hinge, in the sense that a continuous hinge is never added to the catalog. The user, by recognizing the novelty of the continuous hinge concept and adding it to the catalog, performs the actual invention. EDISON relies on a variety of problem solving and naive physics techniques in attempting to salvage "near miss" designs generated by the mutation of topologies. Such techniques comprise entire research areas in themselves.

3.3 Routine

The difference between innovation and routine design is essentially the difference between infinite and finite topologies. While innovative design systems have the potential to generate an infinite number of topologies, routine design systems either generate a finite number of topologies or select from a finite set of predetermined topologies. In either case, the size of any routine design topology is finite. Research taking place in routine design, relating to both transition activity and form activity, is now detailed.

3.3.1 Transition Design Activity

Most of the research taking place in routine design is classified as transitional activity in either the forward (function-to-form) or backward (form-to-function) direction.

The "novel combination" [35], developed by Ulrich and Seering, is a methodology for creating new devices by combining structural attributes of existing devices. Structural entries and associated functional attributes of existing devices are combined into a function/structure topology. The user then inputs a set of functional requirements, which a "novel combination" is generated by replacing the functional entries in the function/structure topology with structural entries. Ulrich and Seering display the utility of this methodology in the design of new fasteners, using structural attributes of existing fasteners as catalog entries.

A limitation of the novel combination approach is that it cannot deal with function sharing, where one structure satisfies several functions -an often essential characteristic of efficient mechanical designs. Ulrich and Seering [36] address this by describing a process for transforming a functional topology into an efficient structural topology. The transformation is intended to map several functional attributes (functional catalog entries) into a single structural attribute (structural catalog entry), resulting in a feasible structural topology. Function sharing is then used to try to eliminate structural entries, altering the functional attributes of the remaining structural entries to retain overall device functionality. The function sharing process is demonstrated for mechanical devices whose behavior can be described by a differential equation relating an input to an output parameter.

Brown and Chandrasekaran [5] discuss an "expert design consultant" that mimics knowledge structures and problem-solving strategies employed by a group of designers. This routine design system is demonstrated within the domain of small table design. The consultant's knowledge structure is a hierarchy of small knowledge sources, called specialists. The problem-solving methodology calls for each specialist to respond to an inquiry, or to seek assistance from a specialist below it. Because the hierarchy of specialists is predetermined, this methodology appears procedural. However, because specialists fail, creating a need for backtracking, the "expert design consultant" is classified here as a routine design system.

Two papers reviewed present work in the area of mechanical linkage generation. Dwell Expert [23] is a system for designing multi-link dwell mechanisms. The system database includes a set of 32 pre-defined mechanism topologies. These topologies are generated by transforming existing linkage information into a form more suitable for an expert system. Using a list of motion requirements and weighting factors indicating the importance of each requirement, Dwell Expert chooses the best fit topology from the list of pre-defined topologies. The resulting mechanism is fine-tuned by refining each catalog entry in the topology. A second linkage paper [17] discusses a system that generates four-bar linkages, categorizing each according to the shapes of the curve and timing path. After the system is taught a number of linkage designs, it chooses the most suitable category to fit curve path and timing path requirements input by the user. This system performs routine design as the topologies are all fixed; that is, they are all four-bar linkages.

A system called PROMPT [27] deals with the same torsional beam problem discussed earlier [6]. The system uses a set of precompiled topologies, called "a graph of models", to convert between various forms of the beam. Using these precompiled topologies, it then simulates innovative design using only routine design features. Within a given topology, only forms are altered making PROMPT appear to belong under the form activity. However, conversion between topologies is in response to functional specifications, causing the system to be classified as transitional. This dual classification is characteristic of many of the reviewed works. It is unclear from the narrow application domain described how the system can be extended to other domains. Clearly, the system is limited by problem size, with the graph of models growing combinatorially as additional topology changes are precompiled.

PRIDE (Pinch Roll transport Interactive Design environment/Expert) [26], performs routine design of paper transport systems. PRIDE's main function is to convert functional input descriptions into structural topologies. These topologies are limited to pinch roller transport systems utilizing known components (catalog entries). While the number of pinch roller pairs is allowed to vary, the variation is limited to a finite set of fixed topologies. However, the complexity of the underlying domain approaches actual design practice.

3.3.2 Form Design Activity

Arciszewski and Aktan [2] discuss a design methodology called "morphological analysis". This methodology is best applied to design problems that can be subdivided into independent subproblems. Solutions to each subproblem are then generated without regard to other subproblems. Potential solutions to the overall problem are taken as combinations of subproblem solutions. Morphological analysis is routine design where the topology is a structure of subproblems, and slots in the topology are filled with potential solutions to each subproblem.

Ward and Seering's "mechanical design compiler" [38,39] uses a catalog of artifacts in the form of mechanical power transmission components to fill in a user-defined fixed topology. As each slot in the topology is filled with a catalog part number, the form of the design is further refined. The completely filled topology represents a physical system satisfying the design requirements. By containing only physical artifacts, Ward and Seering's "catalog" is a subset of the term catalog defined in the vocabulary here.

CASE (Computer Aided Simultaneous Engineering) [31] uses a variety of knowledge based "agents" and "critics" to design an automobile window lift system. The relationships between system components (sector gear, pinion, lift arm, etc.) are determined ahead of time, fixing the topology. Slots in the topology are filled with material properties and dimensional information about each component. Some of this information is provided by lower level procedures such as finite element analysis. The "agents" and "critics" of CASE are based on coded expert knowledge. CASE groups this knowledge around different disciplines, such as assembly, rather than around the product being designed.

3.4 Procedural

Procedural design represents a particular type of algorithmic computation in the design process. To be considered procedural, an algorithm must proceed in a single direction without backtracking. Examples of procedural design routines can be gleaned out of engineering design textbooks, although they may not be presented explicitly as such. Furthermore, such textbook design procedures may in fact contain iterations involving backtracking, again not explicitly presented as such. Shigley and Mitchell [33] or Juvinall [22] are typical textbooks providing information for the design of such items as springs, journal bearings, and shafts. Not all "design procedures" therein represent procedural design in the vocabulary. One clear example of procedural design is the "parametric optimization procedure" in Papalambros and Wilde [29].

Design procedures may vary in level of detail, from procedures aimed at obtaining "ballpark" figures to procedures performing thorough analysis. While many design procedures are transitional, converting functional requirements into the form of an artifact, procedures also exist within the function and form activities. Because Type 4 design procedures are often created as support tools for Type 1 through Type 3 design systems, details of their implementation are often omitted from published literature. For this reason, a review of such works is not attempted here.

An important subset of Type 4 design procedures are those procedures which guarantee optimal results, relative to some predetermined criteria. Such design procedures progress to the optimum without the search or iteration required by the other types of design. Such optimal design procedures are rare in actual practice, and are usually related to monotonic behavior of the underlying mathematical functions with respect to the design variables, e.g., the parametric optimization procedure [29]. For interesting work aimed at generating such procedures based on monotonicity analysis, see Li and Papalambros [25], Agogino and Almgren [1], Choy and Agogino [8], and Rao and Papalambros [30].

3.5 Design Support Tools

As one might suspect, not all of the work taking place in the field of mechanical design automation fits neatly into the classification framework. This realization is due, in part, to the fact that some work is applicable throughout several design types, and therefore cannot be placed into a single classification. However, this shortcoming of the classification framework does not prevent the representation of this research using terms in the vocabulary. Thus, analysis tools, interactive design tools, and optimization tools are very briefly discussed without placing them in the framework.

3.5.1 Analysis Tools

On the surface, analysis tools appear much like the design procedures discussed above. However, whereas design procedures provide predetermined ways for filling topologies, analysis tools do not. Rather, analysis tools operate on filled topologies, returning results which may be used to update or change previous design decisions. To illustrate this, some design-oriented analysis tools are discussed.

Carney and Brown [7] developed a method for determining if two components can be joined in a plug-and-socket manner. Compatibility is determined by a five step process requiring components, described as combinations of polyhedra, to pass increasingly strict criteria. This design analysis tool would typically be accessed during Type 2 or Type 3 design. That is, when a system is choosing catalog entry placements within a topology, the system could access this analysis tool to determine the compatibility of an entry with neighboring entries.

A method called "Design Compatibility Analysis" (DCA) [19] is a tool for the calculation of a "match index". The match index measures the compatibility of components within a system, as well as the compatibility between these same components and system specifications. After assigning an importance rating to each slot of a fixed topology, the user performs Type 3 design by selecting catalog entries to fill these slots. These selections are then analyzed using DCA, which provides feedback indicating components that are major contributors to incompatibility. Some motivation for the method appears to stem also from monotonicity analysis [29]. The application domain is a power generation plant configuration. The topology is filled with various forms, including a boiler, a turbine, and a cooling system. The DCA approach should work equally well filling a functional topology.

3.5.2 Interactive Design Tools

Interactive systems are those design systems that assist a human designer by interacting in a useful manner, such as by making optimal suggestions. They may automate certain aspects of any design activity at any Design Type, explicitly accounting for an interface with a user. The required human-machine interaction study is a major element in the success of such systems.

Jakiela et. al. [20] described an interactive system that provides direct graphical suggestions about structural topology to a human designer working on a traditional CAD system. These suggestions are generated by a background consultation program that evaluates the changes in structural topology resulting from the addition of geometric catalog entries. The suggestions are derived by comparing generated topologies to optimal topologies computed based on the Boothroyd assembly charts [3]. The user has the option of either implementing or ignoring the suggested topological changes. This suggestive system is limited to the domain of geometry-based design, such as the assembly decisions addressed in the Boothroyd charts.

The "First Cut" design support system [9] aids a user performing concurrent product and process design, providing feedback as the designer makes actual design decisions. First Cut operates in various "manufacturing modes," including machining, assembly, and injection molding. These modes deal mainly with the form of the artifact. The scope of the advice offered by First Cut is limited by the ability to code existing human knowledge into usable forms. First Cut places no restrictions on artifact topologies, allowing it to be used to perform innovative design.

3.5.3 Optimization Tools

If design decisions can be made at any level of activity or Design Type, then it is implicitly assumed that there exist several alternatives and criteria for evaluating them. Under this assumption, designing contains a decision-making process and optimization techniques may be applied. These techniques require a formal statement of the design problem, usually in the form of a mathematical model. There is no explicit limitation as to the application of optimization methods for any Design Type. However, appropriate mathematical models may be formulated more easily for Type 3 and Type 4 Design. There is a vast literature in this area which will not be reviewed here. It should be mentioned, however, that formal optimization methods can be used even for Type 1 or Type 2 Design, and most effort in this direction involves more fundamental modeling capabilities.

4 SUMMARY

A design vocabulary was introduced to assist in the description and comparison of diverse research efforts taking place in mechanical design automation. This vocabulary evolved from a particular model of the design process, a model that casts design as an iterative process of combining database entries in meaningful topological combinations. The pervasive use of this model in mechanical design research makes it a suitable candidate as the basis for a standard design vocabulary.

The design vocabulary was partitioned into two major sections: design type and design activity. The vocabulary supporting design type describes the level of invention a design system achieves. The four categories of design type include, in order of decreasing inventiveness: invention, innovation, routine, and procedural design. The vocabulary supporting design activity describes the degree to which a design system deals with functional or structural representations of database entries. The three categories of design activity include function, transition, and form.

The versatility and usefulness of the design vocabulary is demonstrated by reviewing current work in mechanical design automation using the terms of the vocabulary. The research review is structured to represent the classification framework: a two-dimensional matrix with design type and design activity axes. Placement of current work in this structure allows comparison of similar work, as well as identification of both densely and sparsely populated areas of research.

5 BIBLIOGRAPHY

1. Agogino, A.M., and Almgren, A.S., 1987, "Techniques for Integrating Qualitative Reasoning and Symbolic Computation in Engineering Optimization," *Engineering Optimization,* Vol. 12, pp. 117-135.

2. Arciszewski, T., and Aktan, H., 1987, "Innovative Design Method," *Report to the National Science Foundation, Design Theory and Methodology Program,* Wayne State University, Detroit, MI.

3. Boothroyd, G., and Dewhurst, P., 1983, *Design for Assembly - A Designer's Handbook,* Department of Mechanical Engineering, University of Massachusetts, Amherst, Massachusetts.

4. Berg Catalog #B8, 1988, Winfred M. Berg, Inc., East Rockaway, NY.

5. Brown, D.C., and Chandrasekaran, B., 1983, "An Approach to Expert Systems for Mechanical Design," *in Proceedings of the IEEE Computer Society, Trends and Applications '83,* Gaithersburg, MD, pp. 173-180.

6. Cagan, J., and Agogino, A.M., 1987, "Innovative Design of Mechanical Structures from First Principles," *(AI EDAM),* Vol. 1, No. 3, pp. 169-189.

7. Carney, S.P., and Brown, D.C., 1989, "A Qualitative Model for Reasoning about Shape and Fit," *in Proceedings of the AIENG '89 Conference,* Cambridge, U.K.

8. Choy, J. K., and Agogino, A. M., 1986, "SYMON: Automated SYMbolic MONotonicity Analysis System for Qualitative Design Optimization," *in Proceedings, ASME 1986 International Computers in Engineering Conference,* Chicago, IL, pp. 305-310.

9. Cutkosky, M.R., and Tenenbaum, J.M., 1989, "A Methodology and Computational Framework for Concurrent Product and Process Design," *to appear in Mechanism and Machine Theory.*

10. Dixon, J.R., Duffey, M.R., Irani, R.K., Meunier, K.L., and Orelup, M.F., 1988, "A Proposed Taxonomy of Mechanical Design Problems," *in Proceedings of the 1988 International Computers in Engineering Conference*, San Francisco, CA, pp. 41-46.

11. Doyle, R.J., 1988, "Hypothesizing Device Mechanisms: Opening Up the Black Box, " *MIT Artificial Intelligence Laboratory Technical Report 1047,* Massachusetts Institute of Technology, Cambridge, MA.

12. Duffey, M.R., and Dixon, J.R., 1988, "Automating the Design of Extrusions: A Case Study in Geometric and Topological Reasoning for Mechanical Design," *in Proceedings of the ASME Computer in Engineering Conference,* San Francisco, CA.

13. Dyer, M., and Flowers, M., 1984, "Automating Design Invention," *in Proceedings of the Autofact 6 Conference,* Anaheim, CA, pp. 25-1 to 25-21.

14. Dyer, M.G., Flowers, M., and Hodges, J., 1986, "EDISON: An Engineering Design Invention System Operating Naively," *Artificial Intelligence,* Vol. 1, No. 1, pp. 36-44.

15. Freeman, P., and Newell, A., 1971, "A Model for Functional Reasoning in Design," *in Proceedings of the Second IJCAI,* London, England, pp. 621-633.

16. Gero, J.S., and Maher, M.L., 1988, "Designing With Prototypes," *in Proceedings of the International Conference on Engineering Design (ICED '88),* Budapest, Hungary, pp. 74-81.

17. Hoeltzel, D.A., and Chieng, W., 1989, "Pattern Matching Synthesis as an Automated Approach to Mechanism Design," *in Proceedings of the 1989 ASME Design Technical Conferences - 15th Design Automation Conference*, Montreal, Quebec, Vol. 1, pp. 243-250.

18. Hubka, V., 1982, *Principles of Engineering Design*, Butterworth Publishers, London.

19. Ishii, K., and Barkan, P., 1987, "Design Compatibility Analysis - A Framework for Expert Systems in Mechanical System Design," *in Proceedings of the 1987 ASME International Computers in Engineering Conference and Exhibition,* New York, N.Y., Vol. 1, pp. 95-102.

20. Jakiela, M., Papalambros, P.Y., and Ulsoy, A.G., 1985, "Programming Optimal Suggestions in the Design Concept Phase: Application to the Boothroyd Assembly Charts," *Journal of Mechanisms, Transmissions, and Automation in Design,* Vol. 107, pp. 285-291.

21. Joskowicz, L., and Addanki, S., 1988, "From Kinematics to Shape: An Approach to Innovative Design," *in Proceedings of the AAAI Seventh National Conference on Artificial Intelligence,* St. Paul, Minnesota, Vol. 1, pp. 347-352.

22. Juvinall, R.C., 1983, *Fundamentals of Machine Component Design*, John Wiley &Sons, NY.

23. Kota, S., Erdman, A.G., Riley, D.R., Esterline, A., and Slagle, J.R., 1988, "A Network Based Expert System for Intelligent Design of Mechanisms," *(AI EDAM),* Vol. 2, No. 1, pp. 17-32.

24. Lenat, D.B., 1983, "EURISKO: A Program That Learns New Heuristics and Domain Concepts," *Artificial Intelligence,* Vol. 21, pp. 61-98.

25. Li, H.L., and Papalambros, P.Y., 1988, "A Combined Local-Global Active Set Strategy for Nonlinear Design Optimization," *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 110, No. 4, pp. 464-471.

26. Mittal, S., Dym, C.L., and Morjaria, M., 1986, "PRIDE: An Expert System for the Design of Paper Handling Systems," *IEEE Computer,* pp. 102-114.

27. Murthy, S.S., and Addanki, S., 1987, "PROMPT: An Innovative Design Tool," *in Proceedings of the AAAI Sixth National Conference on Artificial Intelligence,* Seattle, WA, Vol. 2, pp. 637-642.

28. Pahl, G. and Beitz, W., 1984, *Engineering Design*, Springer-Verlag, London.

29. Papalambros, P.Y., and Wilde, D.J., 1988, *Principles of Optimal Design: Modelling and Computation*, Cambridge University Press, NY.

30. Rao, J.R., and Papalambros, P.Y., 1987, "Implementation of Semi-Heuristic Reasoning for Boundedness Analysis of Design Optimization Models," *in Proceedings of Advances in Design Automation - 1987*, ASME, NY.

31. Rehg, J., Elfes, A., Talukdar, S., Woodbury, R., Eisenberger, M., and Edahl, R., 1988, "CASE: Computer-Aided Simultaneous Engineering," *C.M.U. Engineering Design Research Center Report EDRC-18-05-88,* Carnegie Mellon University, Pittsburgh, PA.

32. Rinderle, J.R., and Watton, J.D., 1987, "Automatic Identification of Critical Design Relationships," *in Proceedings of the International Conference on Engineering Design (ICED '87),* Cambridge, MA.

33. Shigley, J.E., and Mitchell, L.D., 1983, *Mechanical Engineering Design, Fourth Edition,* McGraw-Hill, Inc., New York.

34. Ullman, D.G., 1989, "A Taxonomy of Mechanical Design", *in Proceedings of the 1989 ASME Design Technical Conferences - 1st International Conference on Design Theory and Methodology*, Montreal, Quebec, Vol. 17, pp. 23-36.

35. Ulrich, K., and Seering, W., 1988, "Computation and Conceptual Design," *Robotics & Computer-Integrated Manufacturing,* Vol. 4, No. 3/4, pp. 309-315.

36. Ulrich, K.T., and Seering, W.P., 1988, "Function Sharing in Mechanical Design, " *in Proceedings of the AAAI Seventh National Conference on Artificial Intelligence,* St. Paul, Minnesota, Vol. 1, pp. 342-346.

37. Ulrich, K.T., and Seering, W.P., 1989, "Synthesis of Schematic Descriptions in Mechanical Design," *Research in Engineering Design,* Vol. 1, No. 1, pp. 3-18.

38. Ward, A.C., and Seering, W.P., 1989, "The Performance of a Mechanical Design "Compiler"," *MIT Artificial Intelligence Laboratory AI Memo No. 1084,* Massachusetts Institute of Technology, Cambridge, MA.

39. Ward, A.C., and Seering, W.P., 1989, "Quantitative Inference in a Mechanical Design "Compiler"," *Artificial Intelligence Laboratory AI Memo No. 1062,* Massachusetts Institute of Technology, Cambridge, MA.