Function-Based Synthesis Methods in Engineering Design: State-of-the-Art, Methods Analysis, and Visions for the Future

Kristin L. Wood* and James L. Greer**
***Department of Mechanical Engineering, The University of Texas, Austin, TX 78712**
****Department of Engineering Mechanics, The United States Air Force Academy**

Abstract

Concept generation is at the heart of engineering design. This chapter considers an emerging tool set for generating concepts: function-based synthesis methods. To understand this tool set, the prominent methods in the field are reviewed and summarized, with a subset being investigated with more technical rigor. In addition to this review and investigation, the methods are analyzed against three models: a method architecture, design process, and research model. This analysis extracts the fundamental features of the methods, provides a basis for comparison, and elicits future research opportunities, directions, and industrial applications. Through this analysis, a clear picture emerges of the function-based synthesis field: Many fundamental research results have been realized, and it is just a matter of time before we have tools to assist product development teams in generating dynamic systems, kinematic structures, and the skeletal backbones of consumer products.

I. OVERTURE: BACKGROUND, INTRODUCTION, AND MOTIVATION

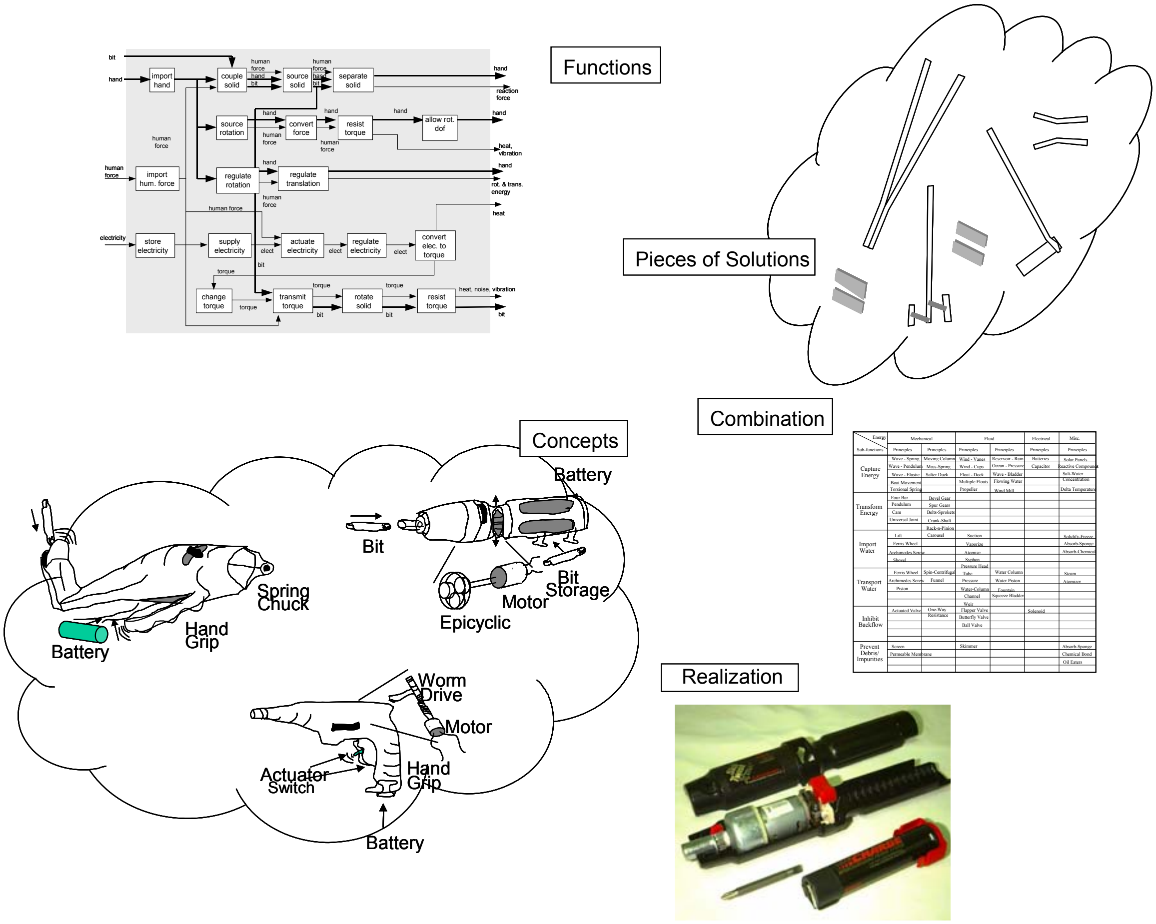


Figure 1. Motivation: A snapshot of functional synthesis during product development. The product example is a cordless, power screwdriver.

Motivation

The activity of concept generation is one of the lampposts of engineering design. It provides a forum for designers to apply creativity and contribute their personal flair. It also represents the time when technology is chosen or developed to fulfill the customer needs.

The imaginary clay of product development is molded during concept generation. Until recently, the tools to shape this clay relied, almost entirely, on the experience and innate abilities of the designer or design team. Concept generation, fundamentally, was considered art not science, informal not formal.

In the last two to three decades, our tool set has changed significantly. Methods are continuously being developed, tested, implemented in industry, and taught to our engineering community (Otto and Wood, 2000). Figure 1 shows a simplistic view of these methods, where customer needs are first transformed to a repeatable functional representation, then to layouts and solution pieces, then to broad combinations and alternative products, and finally to an embodied realization that we can produce for the customer. Based on these methods, our resulting abilities to develop products, and their underlying architectures, are significantly enhanced.

But is this developing tool set complete and convergent? New methods over the last decade (or less) resoundingly indicate a "no" to this question. Formal methods in engineering synthesis, while in their infancy, are emerging as new and *complimentary* possibilities for shaping the clay of product concepts. We see possibilities of formalizing the once thought to be informal, of systematizing the once thought to be purely artistic, and of understanding the once labeled as innate creativity.

This book seeks to convey and advance this tool set, referred to as "formal engineering design synthesis." In this chapter, an important subset of this tool set is considered: function-based synthesis methods.

Background and Issues

To begin the study, let's define important terminology in function-based synthesis methods. By "function," we mean *what* a product or device must do, not *how* it will do it. The concepts of function and behavior are symbiotic; a function is the "what" for a product, and behavior results from how a function is implemented. In terms of modeling or representation, function corresponds to the *action* of a product on its inputs (materials, energies, or signals) to produce desired outputs (Otto and Wood, 2000; Stone and Wood, 1999), such as "convert torque" or "transmit electricity." Recent research in functional modeling has produced formal methods for representing product function as a vocabulary and its corresponding topology of inputs and outputs (McAdams, Stone, and Wood, 1999; Stone and Wood, 1999; Stone, Wood, and Crawford, 2000a,b).

By synthesis, we mean the composition of fundamental or "atomic" elements into combinations that produce unique and desired results. The corresponding function-based synthesis process first combines functional elements, followed by structural and topological elements. The resulting combinations are realizable alternative concepts to solve a design problem.

The emphasis in this chapter is not just on function-based synthesis (as a human activity), but on the formalization of the process. By formal, we mean that the process is founded in a theory, set of theories, or set of principles. Formal function-based synthesis seeks to produce innovative solutions, guided by these theories and principles. Based on the formalization, the process may be coded, at least significantly, as languages, computational methods, and control strategies. Referring to Figure 1, the synthesis process seeks to generate, computationally, significant portions of the concepts and realization, complimenting the skills of a designer or design team.

Objective and Roadmap

The terminology of function-based synthesis aids us in understanding a vision for the field. As a vision, it is not complete or fully realized; yet recent advancements demonstrate its vast potential. The objective of this chapter is to summarize and analyze the current state-of-the-art in function-based synthesis. Through this analysis, we hope to elicit the research strengths, shortcomings, and future direction of the field.

As a roadmap, we first construct our approach for analyzing function-based synthesis methods. Three models are developed to understand and compare the current methods. After developing these models, the field is summarized and segregated into distinct areas. For each area, seminal works are encapsulated, followed by brief analyses with the models. The chapter then presents the technical approach of two representative methods, followed by a discussion of the findings and a subsequent section devoted to modest visions for the future.

II. PREAMBLE: ANALYSIS MODELS

In this section, we present the skeletal structure of our study of functional-based synthesis. Each of the contributions to this field of study could be merely summarized, reporting the basic method and the researchers' opinions regarding the inherent advantages and limitations of the method. Alternatively, we might "step back" from the methods, analyzing the basic contributions, but with a common basis of comparison.

This latter approach is adopted here. Three models are developed below to assist in the analysis: a method architectural model, a design process model, and a research model. These models may be directly compared to the research reported in the literature. Through these comparisons, a number of fundamental questions may be addressed, such as the scope of a method, its coverage or niche in the design or product development process, the need that drove its creation, and its “distance” from application in industrial design practice. By answering such questions, we hope to reveal, objectively, a snapshot of the historical development of the methods and the current state-of-the-art, as well as the avenues for future maturation of the field.

Method Architecture Model

The area of function-based synthesis may be abstracted in terms of a general architecture representing the various generative methods, their inputs/outputs, their fundamental actions, and their layout. Figure 2 shows an architecture for the synthesis methods reviewed in this chapter. This architectural model enables the analysis of the methods, including their similarities and differences. For example, considering Figure 1, a function-based synthesis technique might begin with a functional description of a product opportunity, as shown in the network of functions at the top of the figure. The method may then use an exhaustive search strategy of a database or repository to create a list of potential piece-wise solutions to the functions. A control strategy for combining solutions may then be adopted, followed by the generation of geometry to create solutions (the alternative cordless screwdriver designs shown in Figure 1). These solutions may then be embodied manually or semi-automatically using optimization and/or engineering modeling approaches. Together, these steps, with the requisite theory and implementation, form a function-based synthesis technique. Such a technique may be abstracted and compared to other techniques, in addition to the generic approach of Figure 2, to infer the comprehensiveness and depth of the technique.

Building on this brief example, the architectural model shown in Figure 2 includes four fundamental elements: inputs and outputs, actions or transformations performed by the method, sequential flow from one action to the next, and parallel flow representing alternative paths for a method or hybrid paths for design generation. Inputs for the method architecture begin at the fundamental level of customer needs. These needs are then expressed as either functional specifications or a functional description (functional language), initiating the generation process. Other types of inputs include abstract functional elements and structural elements. Functional elements are abstract representations of designs that convert inputs to outputs. These elements are the lexicons for generating alternative designs, as well as the basis for developing rules for combining elements into patterns that solve the input functional specifications or functional descriptions. Structural elements, alternatively, replace the functional elements to transform the abstraction into realizable forms that include spatial information, such as geometry, topology, orientation, and position, and other characteristics, such as power ratings, etc.

One type of primary output emerges from the architectural model, i.e., generated, alternative designs that satisfy the input specifications. Ideally, based on the model, these designs are fully enumerated in their embodiment. They satisfy the customer needs (feasibility), they have realizable physical forms (design for manufacturing and assembly), their parameters are fully specified, and they have been optimized according to metrics on the input-output criteria. Other types of outputs from the model may occur at any phase of the generation process, depending on the current intent and development status of a given function-based synthesis method. As an example, abstract representations of designs may be generated and output after synthesis of the functional elements has occurred. This type of output is true of any of the synthesis, search, and evaluation actions shown in the architectural model.

Besides inputs and outputs, the core of the model includes actions that transform inputs to relevant outputs in a given generative algorithm. For the model shown in Figure 2, the actions begin with representing the input customer needs and specifications as functional requirements or a functional language. A general workframe for generating designs is then created. This workframe is simply the media or means of representation for the designs to be synthesized, such as data structures in a computational approach and/or a three-dimensional reference frame for the geometric generation of kinematic mechanisms. Abstract functional elements and their associated rules are then created to form the "synthesis engine" for the method. The functional specification or functional-language representation of a design problem is fed to this synthesis engine, resulting in combinations of connected functional elements that satisfy the inputs and generation rules. A search algorithm is implemented to generate these combinations exhaustively or to some termination criterion on the complexity of the functional element chains.

The resulting designs from the synthesis process are evaluated according to their satisfaction of all input criteria (feasibility). This step is followed by a subsequent synthesis of structural elements to create physically realizable designs. Functional elements are systematically replaced with structural elements within a rule-based exhaustive search process. An important action and input into this process is the creation of a catalogue of "atomic" structural elements that satisfy the input-output relationships of the functional elements. Other important actions include the optional creation of topology (position and orientation in the workframe reference axes) and control strategies to initiate, direct, and terminate the synthesis actions.

After synthesizing combinations of structural elements, the combinations are evaluated against the input-output criteria. An optimization method is then employed to refine the structural elements and/or choose designs that are on the Pareto frontier (Otto and Wood, 2000) of the criteria. A subset of the generated designs emerges from this refinement process.

A given function-based synthesis method may include only a subset of the actions shown in Figure 2. Alternative flow paths from input to output (left to right) of the architecture model may thus exist. Alternative paths create the parallel flow structure shown in the figure, such as the creation of functional specifications *or* the creation of a functional description to begin the process. Parallel flows in this model also exist to show cyclic processing, such as the cyclic search for alternative designs in either the synthesis of functional or structural elements.

Overall, the architectural model, Figure 2, illustrates a meta-view of the various function-based synthesis methods. We may use this model to evaluate the characteristics of the methods, such as completeness, type of synthesis, type of input information expected, and the type of strategy used in controlling the generation process, in evaluating the designs, and in optimizing the designs.

Design Process Model

The design process model presented in this chapter is a compilation based on observation from the literature (Iyengar *et al.,* 1994; Hubka & Eder, 1998; Otto and Wood, 2000). In general, the goal of a design process is to synthesize alternative systems that perform the desired functions, meet the performance standards, and satisfy the constraints. In doing so, the design progresses through varying levels of abstraction, from the abstract concept of ascertaining what the customer wants and expects to the embodiment of the final design. At each level, the process is iterative and recursive and achieves incremental progress on a portion of the problem and its ultimate solution.

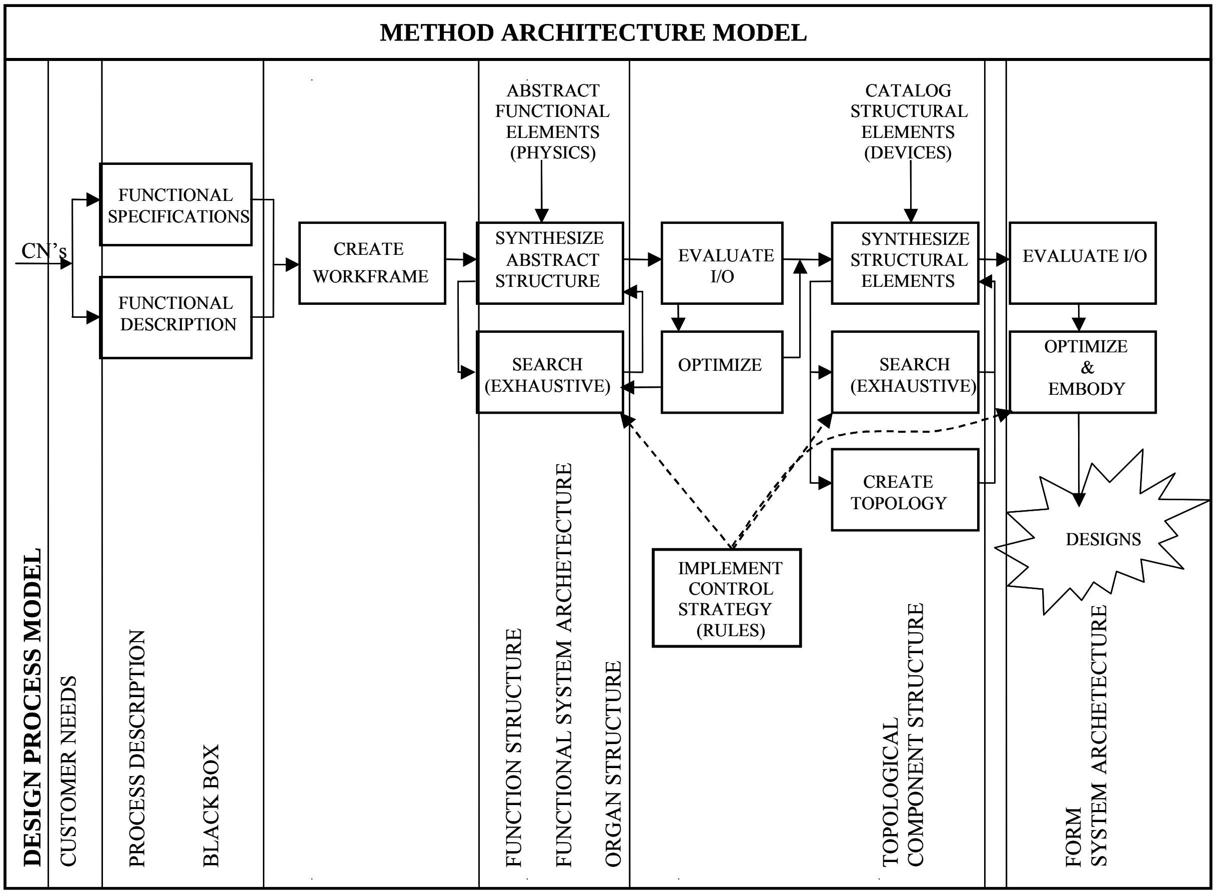


Figure 2. Method Architecture Model, in conjunction with a superimposed Design Process Model.

The design process considered in our analysis is superimposed on the process architecture of Figure 2. The superposition is intended to show how each step of the function-based-synthesis process architecture maps into the overall design process.

The overall process flow in function-based synthesis can be expressed as a hierarchical set of models, beginning with the *customer needs* (CN's), which are a representation of the customer. The CN's model the essence of the interaction between the design artifact and the customer, wherever a customer may lie along the path from manufacturing to end-user.

Based on this set of CN's, the next step is to develop a *process description* for the design. Developing the process description involves analysis of the observed or projected use patterns over the product life cycle of the design, as well as process choices used to facilitate those use patterns; e.g., choosing electric power over internal combustion engine power.

The design progresses to the *black box model*, where specific inputs and outputs to the system are determined based on the process decisions and use patterns previously stated. These inputs and outputs are the major physical flows of the system, and are classified as: energy, material, or signal (Pahl & Beitz, 1996).

The next level of the model is the *functional model (or function structure)* (Hubka, *et al*., 1988; Pahl & Beitz, 1996; Otto & Wood, 1998, 2000; Ullman, 1993), which is a form-independent expression of the product design. The functional model is a domain independent network of functions representing all the necessary working principals needed to carry out the transformation of the physical flows from input to output. These functions are selected from a set of standard functional elements known as basis functions (Stone & Wood, 1997). Each basis function is a primitive element that satisfies the required input/output relationship at a particular node within the network, but is in general energy domain independent.

This network of functions spans the function space such that all input/output requirements are satisfied. The goal at this level is to refine the architecture of the network in search of a preferred functional solution for satisfaction of the CN's. This optimized (or preferred) arrangement is known as the *functional system-architecture*. At this level of abstraction, the system has a specific functional architecture that is capable of carrying out the functional requirements of the design, but has no specific physical embodiment.

Further refinement of the design leads to the *organ structure*, (Hubka and Eder, 1998), where heuristics are used to identify functional elements that can be gathered together to form functional modules (Otto and Wood, 2000; Stone, Wood, and Crawford, 1999). This organ structure is the last level where abstract functional elements are used to describe the artifact. Global product architectural decisions may be made at this point, such as modularity vs. integral architectures. At this point, the design is fully described in terms of functions.

The next level of abstraction associates specific electro-mechanical devices with the basis functions used in the function structure. This level represents the *topological component structure* of the design. Here, structural elements, or devices, are selected to satisfy the input/output requirements at each node in the network. Because these are “real” devices, their interface specifications constrain the system configuration leading to a component topology.

The final level of the model is the *form system-architecture*. This is the least abstract level in the process, where the design is now fully embodied. The goal at this level is to optimize the physical embodiment of the alternative designs that have emerged. It is at this level that the various design methods, such as function integration and design for assembly, are used to optimize the embodiment to satisfy the CN's in the most efficient and effective manner possible with the available resources.

As stated earlier, the design model presented here is a compilation of works by many authors. The goal of presenting it is to use this model as a standard against which each of the methods will be evaluated. We will look at each of the methods with regard to its overall structure and approach to design.

Research Model

The research model presented in this chapter is based on the observation that engineering design research is carried out in both academia and industry, but the driving force behind engineering design research must originate, or at least be targeted, in industry (Cantamessa, 2000). Industry provides the motivation for revision and extension of existing tools and methods as well as the seeds for the germination of new ideas. Development of these tools and methods needs to be nurtured by both academia and industry, but validation must ultimately come from its use in a "live" product development environment. The mechanism for transmission of new and improved methods from research to industry has multiple paths. These paths exist both within industry alone, and between industry and academia by way of university/industry cooperation and by way of the classroom where current research results are incorporated into coursework. Each of the block elements of the proposed research model represents an area of active research in engineering design.

A schematic of the model is presented in Figure 3, where the parenthetical references in several of the elements: Criteria, Description I, Prescription, and Description II, are based on the work of Blessing, *et al.* (1998). These references represent a common basis for understanding design research. The elements of the model are described in more detail in the following paragraphs.

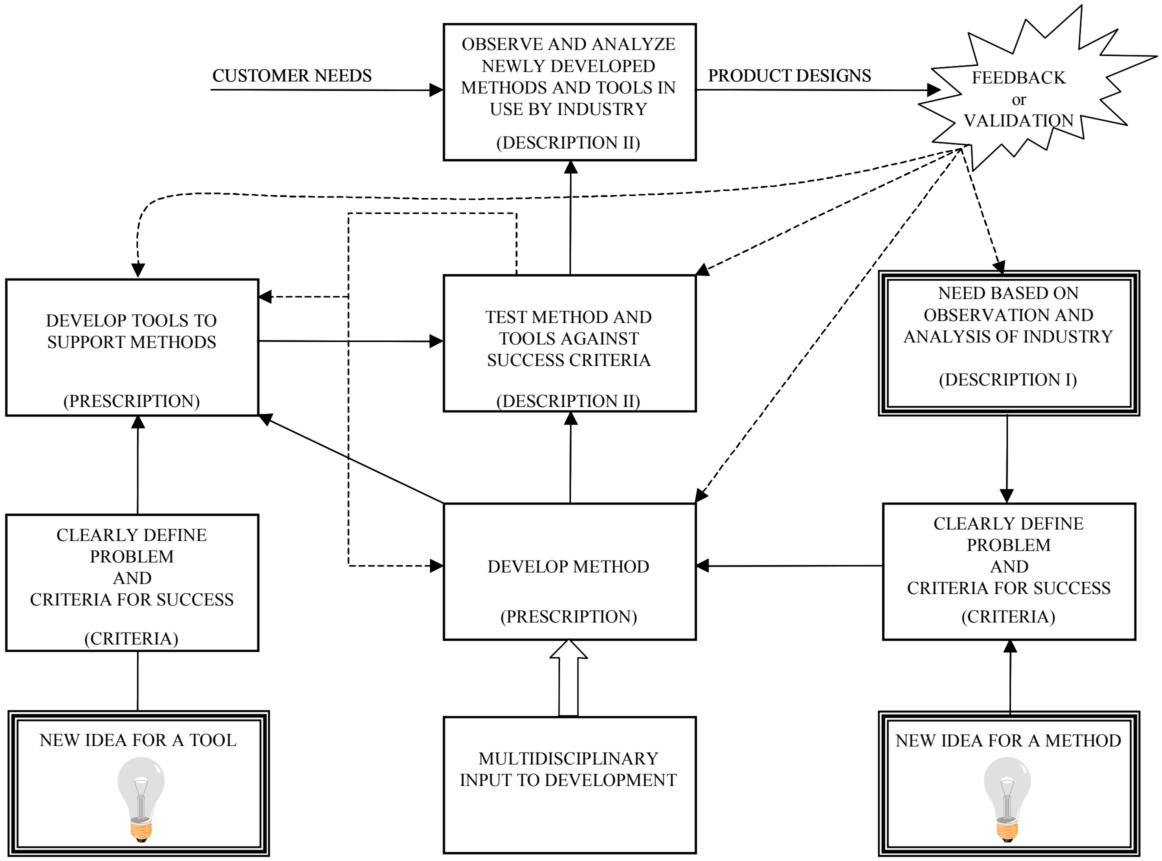


Figure 3. Design Research Model.

The blocks in Figure 3 with bold frames indicate portals of entry to the model. These entry portals represent opportunities for research contributions to the field of function-based design research. Clearly, there are two classes of opportunities, those originating from observation and analysis of industrial practices and processes, and those originating from perceived or expected needs based on original thought (the light bulb). The former are based on descriptive studies of the way product development is done in industry, and the latter are based on intuition, experience, and logical reasoning. In this context, the phrase descriptive study is meant as a study that is in the form of or based upon a coherent theory, not the more common meaning of being vivid or graphic in nature. Both approaches have merits, but it is proposed by Blessing *et al.* (1998), that the descriptive study approach provides the greatest opportunity for dissemination and acceptance of the resulting methods and tools.

Following the flow of an idea from an entry portal to the development element (“Develop Method” or “Develop Approach”) indicates that for any concept to be developed into a method or tool, it must pass through the problem definition and criteria generation element. Problem definition is a critical element in the research model. Without clearly defining the problem to be studied, much effort will be expended in tangential directions without performing useful work in the needed direction. While this “exploratory” approach may be acceptable, or even encouraged, in a university laboratory setting, it is an undesirable approach to research in an industrial setting. According to Schregenberger (1998), design methodologies are an integral part of the broad based management strategies used to increase the stakeholder value of an enterprise. In this environment, a directed action approach to design research is more appropriate, and hence the emphasis on clear definitions of the problem and its objectives. The “criteria,” as used in this element, are defined as the performance metrics by which the research will be evaluated. Blessing, *et al.* (1998) identifies two types of criteria used to measure the success of a research project. The industry measure of success is the market impact of the method or tool that results from the research project, while the laboratory measure of success is satisfaction of the technical requirements used to define the research project at the outset. To be effective as measurement tools, the criteria must be objective measures and some difficulty may be encountered in determining relevant metrics for performance measurement (Duffy and O'Donnell, 1998).

The *method-development* element (“Develop Method”) of the model is one of the two primary areas of research in function-based synthesis (tool development is the other). Here we begin to discuss the prescriptive aspects of the engineering design research model. This element is prescriptive in nature, i.e., prescribed by the innovation and thought processes of the researcher, because the outcome is in the form of instructions, heuristics, guidelines, theory, or advice for practice and application of a method. The essence of this element is the synthesis of useful electro-mechanical design methods based on the results of descriptive studies. Note the inclusion of the multidisciplinary element, which feeds directly into development of the method. This aspect of method-development has been highlighted to bring attention to the need for involvement of disciplines other than mechanical engineering in the process of developing methods. The reasons for this comment are based, first of all, on the fact that method-development is, in and of itself, a design effort, where the artifact being designed is related to the creative process of synthesis, and must involve the sciences that are best suited for understanding such efforts. A second reason for involving experts from other disciplines is the fact that the resulting method must be integrated into a much larger enterprise of product development in the business domain. Pahl (1998) highlights a chronology of collaborative efforts in engineering design research. Schregenberger (1998) pleads the case for involving these other disciplines because they provide expertise in areas well outside the domain of engineering, areas such as business and industrial management, cognitive psychology, organizational psychology, and sociology to name a few. Antonsson (1987) highlighted the need for a more scientific approach in engineering design research. Several disciplines have expertise in clinical experiments using the scientific method, which is invaluable in researching the behavior and internal processes of designers who are engaged in synthesis. In order to make the developed methods more usable in engineering practice, there must be an interface between the method and the engineer, that interface is provided by the tools that are developed to support the methods.

Tool development is a prescriptive effort that leads to instructions, directions, worksheets, or procedures relating a design method to its practical application. Full-scale development of tools is typically a commercial venture carried out by software firms and consulting groups, but the seed that leads to commercial exploitation invariably comes from research. There are two possible paths leading to the tool development research element, the first is as an extension of method-development just discussed, the second path is independent of any current method-development research. As an extension of *method-development*, *tool-development* is seen as a continuation of the process where applying the method to engineering practice is the primary consideration. The independent path approach is the result of perceived or expected needs and is based on original thought (the light bulb). This independent path approach is typically based on the recognition or supposition of the need for a new or improved tool that is associated with an existing design method. Once a method and/or tool have been developed, their suitability for use must be evaluated. As such, this path implicitly includes an element of peril, i.e., if the perceived need does not exist, the resulting method will likely not be accepted or used in actual product development. This inherent danger calls for more industry data to validate research efforts, as shown in the model; however, the independent path approach is still needed to cause bifurcations and leaps in the practice of engineering design.

The methods and tools that result from design research must be tested and validated before they are promoted to industry for implementation. The bottom line is that as researchers, we must ensure that the criteria developed based on descriptive analysis of industry data are used to guide the research and thus lead to useful results. The next two elements of the model are associated with the Description II nomenclature proposed by Blessing *et al.* (1998), and represent the testing and validation process for a method or tool. As the *description* moniker suggests, this is a descriptive study element. Blessing *et al.* highlight two principal difficulties with the validation of design methods and tools, they are: "(i) to identify whether the method or tool has the expected effect on the influencing factors that are addressed directly; and (ii) to identify whether this indeed contributes to success."

The *test-method-and-tools-against-success-criteria* element is designed to address the first of the issues highlighted by Blessing *et al.* In this element, the method or tool is tested and evaluated against the success criteria determined at the outset of the research project. A model/tool that successfully satisfies these criteria will proceed to the next level of validation, while an unsuccessful candidate will be evaluated to determine what further research and development is needed to satisfy the success criteria (iteration/feedback).

Ultimately, the goal of engineering design research is to develop methods and tools that contribute to the success of designers in industry. In order to understand the appropriateness and usefulness of the research, the methods and/or tools that result must be observed and evaluated in an industrial setting (Blessing, *et al.,* 1998). The *observe and analyze newly developed methods and tools in use by industry* element is designed to address this issue by validating the results of design research in an industrial setting, with the preferred setting being a "live" design project. The designs that result from this element are evaluated to determine the efficacy of the method or tool used to execute it.

A second level of feedback is initiated at this level. The resulting designs are analyzed, and the results are fed back to several levels of the design research model. This feedback allows researchers to fine tune their efforts toward more effective methods and tools. In addition, a new round of descriptive studies is begun, thus allowing the studies to evolve to higher levels of refinement. The motivation for this entire effort comes from the fact that the customer of any engineering design research effort is the designer who uses the results, and our goal as researchers, and designers of design methods and tools, is to delight the customer (Otto and Wood, 2000). These final two elements are critical to the ultimate success of integrating engineering design research into industry, as the research must satisfy some need and it must be usable in real-world design situations faced by design engineers.

The purpose of presenting an overall research model is to provide a frame of reference for the various works discussed later in the chapter. We identify the entry portal, the location of the research work within the model, and the level of progress toward validating each of the works presented.

III. TENOR: STATE-OF-THE-ART, BODIES OF RESEARCH

Overview of the Research Field

With the method-architecture, design process, and research models defined, this section seeks to analyze the field of function-based synthesis research. To begin this analysis, let's consider a generic overview. As shown in Figure 4, the field may be segregated into a number of focal areas: Synthesis of Dynamic Systems (Bond Graph Chunks, Function-Based Bond Graph Chunks, Impedance Methods); Agent-Based Methods; and Catalogue Design Methods (Component Composition, Set-Based, Grammars). All of these areas share the common goal of generating mechanical or electro-mechanical designs based on an input functional model or functional specification. They also share the common goal of formal synthesis, i.e., developing, ultimately, a mathematical, grammar-based, or lexicon language for transforming functional representations to physical designs or products. In the following sections, we study selected research work from each of the areas shown in Figure 4, applying the models developed above.

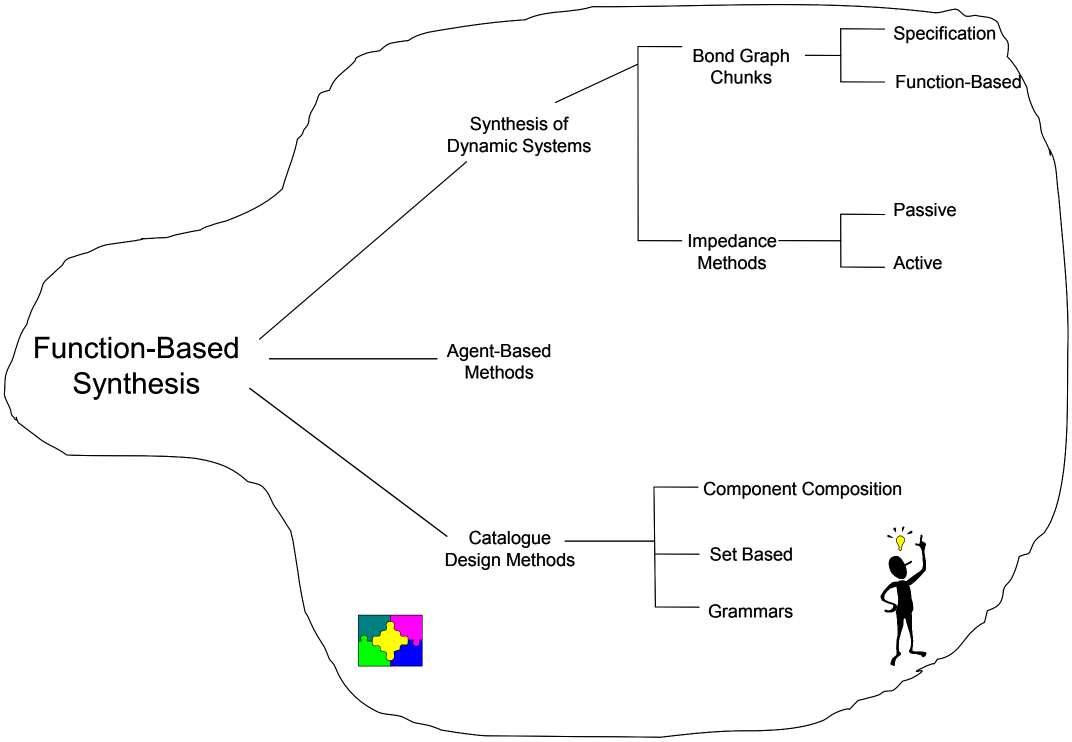


Figure 4. Overview of function-based synthesis research.

Synthesis of Dynamic Systems

The bulk of the research work carried out in synthesizing dynamic systems focuses on the use of bond graph methods (Paynter, 1961; Karnopp *et al.*, 1990), or equivalent, as the formal schematic description of a system being designed. Concentrating on the schematic description without regard to a physical description forces the designer to abstract the functional behavior before worrying about instantiation. Ulrich (1989) defines a schematic description as a "graph of functional elements", and schematic synthesis is defined as "generating a schematic description in response to a specification of desired device behavior." The use of schematic descriptions in function-based synthesis of dynamic systems effectively separates functional issues from structural issues. Hence, synthesis of dynamic systems has come to mean the generation of bond graphs (or equivalent) that represent the functional relationships within a dynamic system.

Bond graphs are non-domain specific representations of the exchange of energy in systems composed of lumped parameter elements. Systems are represented as interconnected components with power flows across their interfaces (ports). The ports are specified in terms of effort and flow variables in various domains. The effort variables are force, torque, voltage, and pressure, for example, while the flow variables are velocities, current, magnetic flux, and volume flow. A distinct advantage of using bond graphs as the schematic description is that the equations of motion are generated as a byproduct of their use. The effort and flow variables can be represented as generalized variables which are non-domain specific, thus allowing the use of this schematic graph to describe multiple energy domains. The use of bond graphs in function-based synthesis has evolved into two distinct fields.

Bond Graph Chunks Methods

The first field is closely related to the work of Ulrich (Ulrich, 1988; Ulrich and Seering, 1989) in which "chunks" (or modules) of bond graphs are used to specify function. The problem initially consists of an input chunk and an output chunk, with the remainder of the system being unknown and represented by a black box. The input and output are specified using three elements: (1) an effort or flow variable for the input and the output, (2) identification of the integral or derivative relation between the input and output variables, and (3) specification of a lumped parameter model of the input and output. The design process progresses by searching the knowledge base of known bond graph chunks while ensuring domain consistency. Synthesis of the unknown black box system is built from these bond graph chunks to form what is known as a "power spine." Configurations of the power spine are exhaustively generated subject to the constraint that no more than three inter-medium transformations exist in the power spine. The most important result of generating candidate designs using the power spine is that the designs are minimal. To be minimal, a candidate design must satisfy the input/output specification and it must not have more than three inter-medium transformations. The approach of Ulrich and Seering has been characterized as a design and debug strategy, and one limitation of the approach is that it only considers SISO (Single-Input Single-Output) systems.

Another work that utilizes the representational power of bond graphs is that of Prabhu and Taylor (Prabhu and Taylor, 1989). Prabhu and Taylor treat the process of designing systems, given functional requirements, as a process of mapping the requirements onto physical artifacts. Functional requirements, shown in Table 1, consist of a set of power, effort, and flow variables that exist in the requirements space. A library of predefined components is used to represent the physical artifacts that exist in the design space. The authors define a vector representation of the functional requirements as a combination of the elements from Table 1, consisting of the following variables:

$i \equiv$ port index
$e \equiv$ effort variable value
$f \equiv$ flow variable value
$\delta \equiv$ 1 if power flow is out of the system
$\equiv$ -1 if power flow is into the system
$D \equiv$ domain (translational, rotational, etc.)

Table 1. Functional requirement specifications.

Ports at which only an effort is specified
Ports at which only a flow is specified
Ports at which only a power-flow is specified
Ports at which element of effort, flow or power is specified (the domain is specified)

The function-based synthesis process begins with this vector specification of the input and output functions attached to a black box representing the unknown system. The next step in the process is to reduce the vector problem to a scalar problem and solve it by selecting a spanning set of functional primitives that satisfies the input-output specifications without regard to power flow direction (Prabhu and Taylor, 1988). The result of the scalar problem solution is a design topology representation of the black box that is bond graph based. This scalar topological solution will in general not satisfy the vector requirements, thus a "vector tuning" operation is implemented to satisfy the magnitude, orientation, and position requirements of the input-output requirement vector.

Analysis – Architecture of "Bond Graph Chunks" Methods: The methods by Ulrich and Seering and Prabhu and Taylor are quite innovative. They use the physics underlying power-flow systems to design a correct abstract schematic, followed by a physical instantiation.

Considering Figure 2, these methods contributed significantly to many of the early activities needed for a fully-developed function-based approach. However, as with the methods reviewed for grammar based conceptual design (Cagan, Chapter X), bond-graph-chunks methods are short of completeness. Customer needs are again implicitly treated as an input leading to the development of a black box model. Input to the method consists of the type of junction, one or zero; the relationship between the input and output, derivative, integral, proportional, etc; and the bond graph chunks that represent the input and output functions. The method synthesizes a set of candidate schematic descriptions from which all possible solutions can be generated, and allows multiple solutions to be evaluated. The number of solutions is controlled by the power spine concept and the limitation of three energy transformations. Beyond this control, the remainder of the control strategy is based in the user of the method. The designer must exercise considerable judgment in selecting which designs will be pursued. There is no optimization scheme *per se*; once again this activity falls to the user.

Analysis – Design Model: The bond-graph-chunks methods use a portion of the fundamental design method that is proposed in this chapter (Figure 2). The method begins at the black box level, and stops just after the synthesis of abstract functional elements. The workframe of the method consists of bond graph schematic synthesis. The method has great potential to be expanded and advanced, especially with respect to coverage of the design process and domains of application.

Analysis – Research Model: The bond graph chunks methods enter the model at the Idea for a New Method portal (Figure 3). There is not a documented need for the method, and its applicability to industry is not expressed. The works reviewed in this section are preliminary and have spawned other research efforts. In this context, the work can be considered fundamental in nature; yet, there is a tremendous need to obtain industry data and develop the methods toward specific industrial sectors.

Function Based Bond-Graph Chunks Methods

The furthest evolution of the bond graph chunks approach to function based synthesis is represented by the work of Malmqvist (1994). The foundation of this work is the technical systems theory of Hubka (Hubka, 1982; Hubka & Eder, 1988), and bond graph theory as previously discussed. Hubka's technical systems theory is summarized as a general characterization of the purpose of machines. A machine can be represented as four separate models: the process structure, the function structure, the organ structure, and the component structure. In addition, an extension of the function vocabulary developed by Krumhauer (1974), Table 2, is used to develop a library of elementary functions. These two fundamental modeling approaches along with the function vocabulary are melded together with the design methodology proposed by Pahl and Beitz (1995) to produce the dynamic systems synthesis method proposed by Malmqvist.

The input to the synthesis procedure is a black box model of the artifact to be designed. Output from the method consists of a set of alternative function structures that correspond to the black box model, and a set of matching organ structures for each alternative function structure. The design problem is specified by the flows at the input and output ports with the unknown system represented by a black box model. The flows are represented as bond graph chunks that are specified by effort and flow variables and are positioned and oriented in space. The specifications are given as: port type (input or output), physical domain, power value, effort or flow, time operator (integral, derivative, constant), position, and orientation.

Once specification of the model is complete, the method synthesizes alternative function structures by first constructing a "minimal" function structure by applying decomposition rules. This function structure is then varied systematically by applying variation rules. In the context of this work, "minimal" has essentially the same meaning as it does for Ulrich and Seering (bond-graph-chunks method above). A minimal function structure consists of a set of elementary functions, any of which if removed would cause the system to no longer perform the required global function. The minimal function structure is generated by searching the library of elementary functions for a minimal set that satisfies the input and output specifications of the black box. The method is equipped to deal with SISO systems with a constant time operator, SISO dynamic systems, and MIMO (Multiple-Input Multiple-Output) systems.

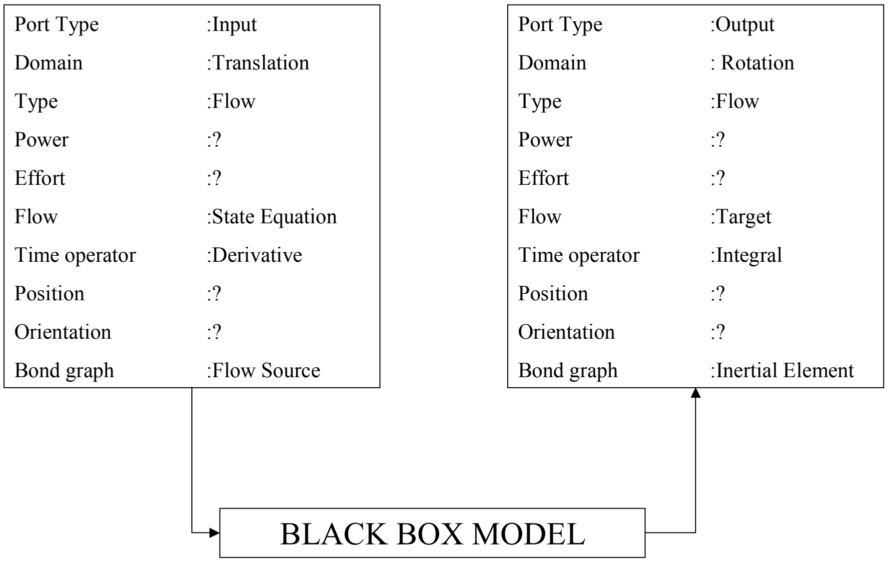
Table 2. Function vocabulary (Malmqvist, 1994).

Elementary Functions	*Functions with Effects*	*Refinement of Functions*	
Change	Change Physical Domain	Rotation → Translation Rotation → Electricity Rotation → Hydraulics : : Translation → Electricity Translation → Hydraulics : :	
	Change Causality	Flow → Effort Effort → Flow	
Vary Magnitude	Non-controlled	Steady State Response	Proportional Integrate Derivate Complex Dependency Frequency Relationships
		Dynamic Response	
	Controlled by signal		
Connect	Join	Join Effort Join Flow	
	Separate	Separate Effort Separate Flow	
Channel	Change Place	Change Position Change Orientation	
	Stop	Translation support Rotation Support Prevent Leakage Isolate	
Store	Store Kinetic Energy	Effort Input Flow Input	
	Store Potential Energy	Effort Input Flow Input	

Synthesis of constant power SISO systems is a straightforward search for elementary functions that satisfy the difference between input and output specifications. For dynamic functions, the straight mapping to a single elementary function does not hold. This difficulty arises because the "requirements on causality and time operators can not be decomposed separately, since physical realizations of dynamic functions always involve two physical effects." (Malmqvist, 1994). The first effect is realized through the use of capacitive or inertial elements, but the side effect of adding an inertial or capacitive element is that the causality of the system is changed. This result can have a positive or negative side effect depending on whether the desired change involves the time operator and the causality, or just the time operator. In either case, techniques are available to exploit or counter the effect. The method treats MIMO systems by either joining or separating functions. The system strategy is to first identify *common* and *individual* functions, and then the *joining* and *separating* functions are inserted between the common and individual functions, respectively. Once a minimal function structure is established, function structure variants are synthesized by transforming the minimal function structure in such a way that the overall specification is preserved. To achieve these various configurations, variation rules are applied.

Organ structures are now created using the results of the function structure variation. To create an organ structure, organs that match the functions within the function structure are selected from the library using a strategy of "one function at a time." The organ structure is constructed such that the overall function of the organ structure matches that of the function structure. Alternative organ structures are generated when more than one organ matches the function in question.

The method is applied to the synthesis of a design for an accelerometer. The input and output functions used to specify the problem are twofold. The input flow will be translation, and the output flow will be rotation of a dial; the time operator of the input will be derivative and the output time operator will be integral, refer to Figure 5. The results are encouraging, as the method produces a solution consisting of a rack and pinion, two grounded torsional springs, two rotational dashpots, and a dial to indicate acceleration. The equations of motion of the synthesized system are then exercised using a smoothly varying function that reaches a steady state value. The result is a set of state equations that produce the desired result, proportionality between an input acceleration and an output displacement.



Port Type	:Input
Domain	:Translation
Type	:Flow
Power	:?
Effort	:?
Flow	:State Equation
Time operator	:Derivative
Position	:?
Orientation	:?
Bond graph	:Flow Source

Port Type	:Output
Domain	: Rotation
Type	:Flow
Power	:?
Effort	:?
Flow	:Target
Time operator	:Integral
Position	:?
Orientation	:?
Bond graph	:Inertial Element

Figure 5. Example Input/Output model for Malmqvist (1994).

<u>*Analysis – Architecture of Function-Based Dynamic Systems Methods*</u>: Overall, the function based bond graph method of Malmqvist (1994) is the most complete approach of all those reviewed in the bond graph chunks class of methods. Input to the model is given as shown in Figure 5. The method begins with a black box model and addresses all aspects of the proposed process architecture up through synthesis of structural elements with topology. The final result is a set of alternative designs containing components with functional and spatial characteristics that satisfy the overall design goal set forth in the initial specification. Control of the generation process is accomplished by constraining the synthesized function structure to be minimal. The generation process is guided by the decomposition and variation rules as well as the criteria for the organ structure search.

<u>*Analysis – Design Model*</u>: The design model for this method is based on similar fundamentals as the design model proposed for this chapter. The primary difference lies in the absence of the customer voice. Final output is at the level of a topological component structure with embodiment of the design represented in a limited way. No optimization of the embodiment is done hence no form system-architecture is achieved. Benefits of this method include the

generation of multiple concept variants with global geometric and functional constraints satisfied.

Analysis – Research Model: Malmqvist's hypothetical basis for the research is "that computational conceptual design systems may be improved by matching the function vocabulary used to the modeling concepts of a physical systems modeling language." (Malmqvist, 1995). Entry to the research model must then be through the New Idea for a Method portal. Method development and tool development are achieved (in part), as is testing against success criteria through an example, where an accelerometer with a known output response is designed. Success is claimed based on the example, with the caveat that considerable work remains before "real" problems can be treated. Regardless of the shortcomings identified, considerable resonance exists between this method and the models proposed in this chapter. Industrial sectors should be sought, with "real" problems identified.

Impedance Methods

The second field of research that has evolved out of bond graph analysis is the use of impedance techniques for the synthesis of dynamic systems. Impedance techniques use many of the same concepts and functional models as the bond graph chunks approach. The system is still specified as a set of input and output functional criteria, with the unknown system lying between the two. The fundamental difference between the methods is the approach to synthesizing the unknown system.

The basis for the impedance method harkens back to the network synthesis work of Foster (1924) who developed the technique and Cauer (1958) who extended the work to apply to RL and RC circuits. The method employs partial fraction expansion to separate the impedance relation until the fractional components are recognizable as representing physical artifacts. According to Redfield and Krishnan (1993), network synthesis is the generation of electrical circuits and networks based on filter specifications in the frequency domain. Early work formulated the impedance synthesis problem in which an impedance relationship between a voltage and current at a single power port is represented as a rational function. Mathematically, impedances are either positive real corresponding to passive systems or negative real corresponding to active systems. This distinction results in two distinct areas of impedance methods research.

Impedance Methods I: Passive Systems

Synthesis of passive systems using impedance bond graph techniques is based on the work of Redfield (1993) and Redfield and Krishnan (1991, 1993). Passive systems are systems that require no active control or external source of power. Network synthesis of *passive* functions typically separates an impedance specification into smaller pieces in a process known as reticulation. Each of the smaller parts is in turn reticulated until the smaller elements are recognizable as representations of physical circuit elements.

The work of Redfield and Krishnan (1993) adopts the network synthesis problem to the bond graph domain and extends it to allow the generation of multiple configurations for a given impedance. In this context, impedance is defined as the ratio of effort and flow at a single port. In addition, impedance is the performance specification for the system. The goal of the work is to synthesize bond graphs and sub bond graphs from impedances, and then associate the bond graphs with physical artifacts. At the fundamental level, this method uses a mapping of reticulated impedances to simple bond graph elements. As an example, Table 3 contains a basic set of bond graph elements and the associated impedances, but it is possible to generate much more sophisticated bond-graph-to-impedance mappings. At a more sophisticated level, this technique can be automated to include libraries of mappings to be accessed during an automated design search process (Redfield & Krishnan, 1991).

Table 3. Bond Graph to Impedance Mappings (basic one-port impedances).

$\xrightarrow[f]{e}\mid$ R	$\frac{f(s)}{e(s)} = \frac{1}{R}$	$\xrightarrow[f]{e}\mid$ C	$\frac{e(s)}{f(s)} = C$	$\xrightarrow[f]{e}\mid$ I	$\frac{f(s)}{e(s)} = \frac{1}{Is}$
$\xrightarrow[f]{e}\mid$ R	$\frac{e(s)}{f(s)} = R$	$\xrightarrow[f]{e}\mid$ C	$\frac{f(s)}{e(s)} = \frac{1}{Cs}$	$\xrightarrow[f]{e}\mid$ I	$\frac{e(s)}{f(s)} = Is$

Synthesis of dynamic systems, using passive elements, is demonstrated in Redfield (1993). In this work, the design of a velocity sensor is synthesized. The functional specification is given as an initial frequency specification for the system. This frequency specification is given as a ratio of the input velocity to some measurable output such as displacement or relative velocity. The schematic description of the system is as shown in Figure 6. In this figure, "D" represents the unknown system in both the bond graph and the schematic.

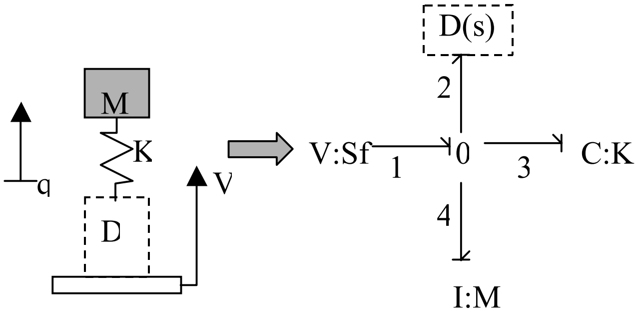


Figure 6. Velocity Indicator schematic and bond graph.

The impedance is given by the ratio of the effort on bond 1 to the flow on bond 1. This impedance can be represented by the transfer function:

$$\frac{F_s(s)}{V(s)} = \frac{e_1(s)}{f_1(s)} = \frac{Is}{CIs^2 + D(s)Is + 1} = A, \tag{1}$$

where "A" represents a performance specification based on the assumption that all frequencies are important, therefore a constant relationship is required for all frequencies. The "s" represents the Laplace operator (or more generally an operator that converts a differential equation to an algebraic equation), and $F_s(s)$, the spring force, is the desired output. This ratio can be solved for *D(s)*, and reticulated to produce the following relationship:

$$D(s) = \frac{Is - CIAs^2 - A}{AIs} = \frac{1}{A} - Cs - \frac{1}{Is}. \tag{2}$$

Using these simple ratios and Table 3, a bond graph can be constructed using the synthesized representation for *D(s)*. This bond graph is the schematic description of the design for a velocity indicator. Clearly, this example illustrates a simplified overview of the process, and much more complex frequency response specifications can be imposed on the system. For further review of this technique, consider Redfield and Krishnan (1993), and Redfield (1993). For a more advanced application of this method, consider Redfield (1999), where the method has been applied to the synthesis of novel design concepts for a constantly variable transmission (CVT).

Analysis – Architecture of Impedance Methods I: Redfield (1993) provides a clear picture of the architecture of this method. Figure 7 illustrates a reproduction of the method as the author envisioned it.

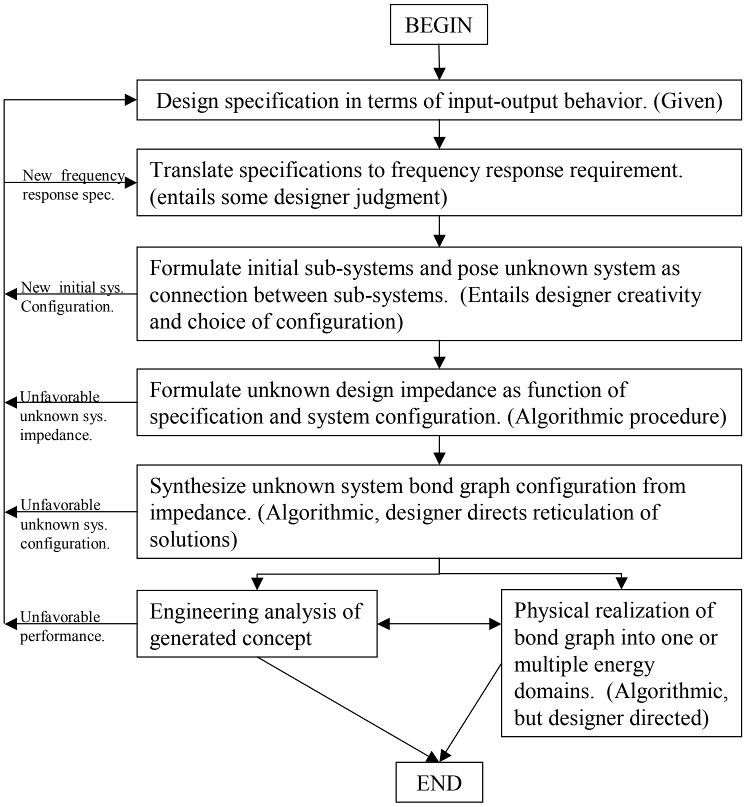


Figure 7. Method architecture of an impedance, dynamic systems synthesis methods (Redfield, 1993).

Clearly, customer needs are assumed to exist, but aren't treated explicitly, rather the input to the method is a design specification. Design specifications are in the form of a desired system response and an unknown impedance representing the system to be designed. Figure 7 also highlights the flow and feedback mechanism of the method. Overall, the architecture is similar to that proposed in this chapter starting with the black box model and proceeding to search and synthesis of abstract structures. The control mechanisms of this method intimately involve the designer who directs the process from concept generation to optimization.

While the method has these features, more automation of the technique is required. The method is algorithmic and procedural in nature, but has yet to be effectively coded as a testbed tool. In turn, industrial applications, such as with the CVT, are currently being pursued; yet, descriptive studies are needed to provide clear conduits to design practice.

Analysis – Design Model: Correlation between the design approach of this method and the one proposed is good for the region of the model covered by the method. Coverage is limited to the region from the black box model to the function system-architecture. Further comparison is not possible because of the limited scope of the method.

Analysis – Research Model: Synthesis of passive dynamic systems is carried out algorithmically, yet in a very manual way using this technique. Clearly this set of works by Redfield and Krishnan (1991, 1993) and Redfield (1993) are foundational in this field with the intention of developing a method, not an automated tool. Entry into the research model is through the Idea for a New Method, and proceeds to the Test Method element. Limited use of success criteria is established, and testing is done at a more conceptual level.

Impedance Methods I: Passive Systems (cont.)

Another notable work in the area of synthesis of passive systems is by Tay, *et al.* (1998). In this work, the authors present a comprehensive approach to design synthesis and the generation of concept variants. The method uses bond graphs as the schematic description of the system, and employs genetic algorithms to provide the operators needed in the evolution of design variants. The program flow chart, Figure 8, gives a general overview of the design method embodied by the program.

The bond graph topology is represented as a two dimensional string called a genome. To formulate the problem, the program combines the genome with a binary-to-decimal string representing the parameter values to form an incidence matrix. Once the problem is formulated, the mutation, selection, and combination operators of the genetic algorithm transform the population of randomly generated two-dimensional binary strings to a valid bond graph with the desired dynamic characteristics. A valid bond graph satisfies the conditions of Kirchhoff's Current and Voltage Laws, geometric compatibility, Newton's Laws, and conservation of energy. In all, three programs are presented: *Topology Generator, Performance Generator, and Variant.* For a more detailed account of this method, refer to the referenced article by Tay, *et al.*

Analysis – Architecture of Tay's Passive Method: Figure 8 is a good representation of the process model used in this method. Input to the model is a black box model, with a genetic algorithm used to direct and control the concept generation process. A sample input vector is as shown in Table 4.

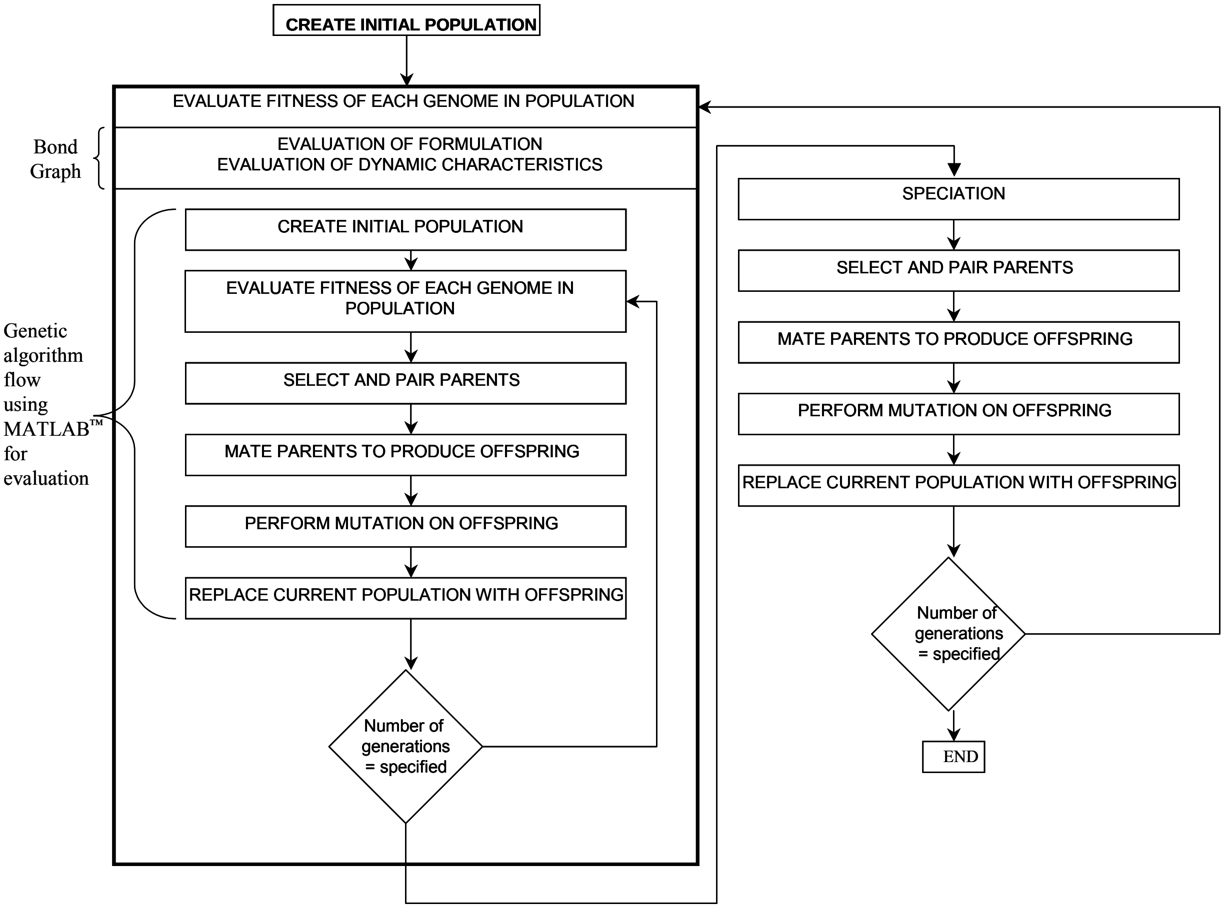


Figure 8. Flow of the synthesis program by Tay, *et al.* (1998).

Table 4. Input quantities for Tay, *et al.* (1998).

MODEL PARAMETERS		**FIRST STAGE GENETIC ALGORITHM CONTROL**
1	Max number of storage elements	Number of generations
2	Max number of dissipative elements	Size of population
3	Range of parameter values	Probability of crossover
4	Nominal parameter values	Probability of mutation
5	Input quantity	Proportion for replacement
6	Output quantity	Sharing function used
		SECOND STAGE INPUTS
		Number of generations
		Size of population
		Probability of crossover
		Probability of mutation

Model parameter inputs are the most interesting elements of the table for this discussion. A note of absence in the input specifications is geometric content. Energy is the only global type parameter being specified, thus allowing the method to generate concepts free of any geometric constraints but forcing the designer to invest significant effort embodying the conceptual results of the method.

Analysis – Design Model: A method for concept generation is developed, and as a method, the work attempts to be comprehensive in nature. There are still omissions to be noted, first of which is the voice of the customer. Additionally, the method stops short of actual embodiment of a design by terminating the process at the functional model variant level. Initial specification of the artifact is again at the black box level, with the generation of function-structure concept variants being carried out by the application of a genetic algorithm.

Analysis – Research Model: Tay, *et al.* describe the research as being "situated comfortably in the computational-conceptual design area." They note its contribution in two areas: (1) it uses operators rather than rules allowing the designer to explore areas with which he/she is not familiar, and (2) it contributes to the application of genetic algorithms to dynamic systems. Synthesis of dynamic systems using bond graph chunks is extended considerably by the inclusion of commercial software in the analysis (MATLAB™) and the inclusion of a genetic algorithmic approach. Although the authors are affiliated with industry, the influence of industry as a prime mover is still missing. Several example designs are generated then compared to known good solutions to test the effectiveness of the model, resulting in novel concepts for a pumping machine that uses components from multiple energy domains.

Impedance Methods II: Active Systems

The limitation of dynamic system synthesis using passive elements is that it cannot generate designs that include active elements, thus restricting the generality of the method. The second classification of dynamic systems synthesis, using impedance methods and bond graphs, expands this approach to include active elements in the schematic description of the design.

The final method to be treated in this section is the synthesis of dynamic systems with active elements. Active elements are elements that require a power source that is external to the system in which they operate. The work of Connolly and Longoria (2000) is representative of work in this area. The method proposed by Connolly and Longoria uses negative impedances, generally a non-realizable physical phenomenon, as a mathematical modeling tool to represent the behavior of active elements. This approach is structurally the same as that presented by Redfield and Krishnan (1993), but the inclusion of negative impedances opens the door to designs with actively controlled elements. At the point in the process where the system is fully reticulated and the resulting impedance relationships are mapped to bond graph form, the system is fully designed from a functional perspective (at least in terms of dynamic behavior functions). At this point, there is still no physical or topological form to the design. Connolly and Longoria propose a four-step process to realize, physically, the active elements.

The first step in physically realizing the active elements is to separate the active from the passive elements in the bond graph structure. In general, all active and passive elements are not separable. The next step is to decide whether the separable elements should be realized individually or if they should be incorporated into the design of the active elements (function sharing). Next, a physical realization of the negative bond graph elements is created. Finally, a control strategy for the active element must be designed. This control strategy is unique to the active system synthesis method. At this point, a tuning phase is entered as the model of the system is tested against the system specifications.

Connolly and Longoria test their synthesis method in an example, where an active vehicle suspension is redesigned. The vehicle suspension model is a quarter vehicle model that includes a tire, an unsprung mass, and a sprung mass. Between the sprung and unsprung masses exists a spring and an unknown suspension system specified to have some impedance relationship $Z(s)$. The input specification to this method is in the form of a transfer function of the desired response, in the form of either experimental data, or it can be prescriptive in nature. Using a bond graph, which represents the system model with the unknown impedance of the suspension system as $Z(s)$, a transfer function for the system is generated based on this bond graph. The specified and generated transfer functions are set equal, and the resulting equality is solved for the unknown impedance. The resulting rational function is then reticulated to find the recognizable impedances that represent physical elements. Embodiment of the design follows this step, with the replacement of these elements with actual components. This last stage is not part of the synthesis procedure/architecture.

Experimental or prescribed data are also used as a basis for evaluation of the performance of the dynamic system synthesized using the method. The authors claim good correspondence between the experimental data and the behavior of the synthesized active suspension.

<u>*Analysis – Architecture of Active Methods and Design Model*</u>: In general, the approach to the design taken in this method is identical to that of the bond-graph-chunks method developed by Redfield and Krishnan (1993). With regard to completeness, this method stops short of emulating the entire method architecture proposed in this chapter. Once again, the method begins at the black box level and carries forward to the synthesis of abstract structures. Some effort is put forth to develop embodiments of the design, but this is done through the experience of the designer and is very much dependant on bond graph analysis experience of the designer. In addition, the designer, with no automated synthesis, carries out control of the process. To the authors' credit, this is a very new approach to dynamic systems synthesis that has not shown up in the literature prior to this work.

With regard to the Design Model (Figure 2), the middle portion of the model is covered from black box model to the organ structure level (in part). Likewise, entry into Research Model (Figure 3) is through the "New Idea for Method" path, with corresponding analysis to that of Redfield and Krishnan.

Agent-Based Methods

Another focal area of function-based synthesis is agent-based methods. Agent-based systems implement computational methods founded in the analogy of basic artificial life elements that are coded with certain reflexive operations. When combined through a control strategy, these elements, or agents, can create what appear to be quite complicated behaviors.

Of particular note in this area is the work by Campbell, Cagan, and Kotovsky (1999, 2000). In this work, a four-prong strategy is adopted for generating concepts to particular design domains. This strategy is embodied, as a computational design method, through what is termed basic subsystems of an A-Design Theory.

The subsystems of this theory are (1) an agent architecture for creating and improving design alternatives, (2) a functional representation of a design problem that is synergistic with the agents, (3) a multi-objective decision-making approach for searching the design space and for retaining or discarding potential solutions, and (4) an algorithmic approach, based on iteration, that seeks to improve generated solutions (Campbell, Cagan, and Kotovsky, 1999). Table 5 illustrates a generic functional specification, known as functional parameters (FP), for electro-mechanical design. This specification language represents the required input-output states of the machine being designed. Table 6 shows the input and output FP for a human weighing machine application.

Table 5. Function parameter (specification} structure and contents for input-output representation (Campbell, *et al*., 2000).

Function Variables (flows/constraints)	*Possible States/Values*
Through	{any real number, bounded, unbounded}
Across-Integral	{any real number, bounded, unbounded}
Across-None	{any real number, bounded, unbounded}
Across-Differential	{any real number, bounded, unbounded}
Class	{power, signal, material}
Domain	{trans, rotate, electric, hydraulic}
Position	{x,y,z}
Orientation	{θ, ϕ, ψ}
Interface	{standard size, e.g., 9/16” bolt}
Direction	{nil, source, sink}

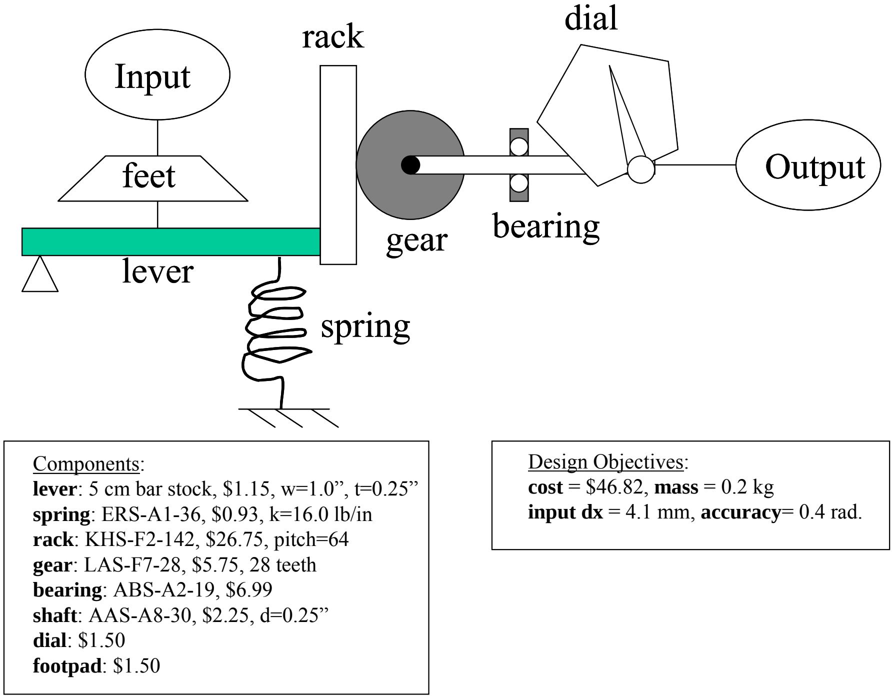
Table 6. Input-Output FP’s for a human weight machine application (Campbell, *et al*., 2000).

Input Function Variables	*Input States/Values*	*Output Function Variables*	*Output States/Values*
Through	{[0 300]}	**Through**	nil
Across-Integral	{goal 0}	**Across-Integral**	{goal [0 5]}
Across-None	{goal bounded}	**Across-None**	{goal bounded}
Across-Differential	nil	**Across-Differential**	nil
Class	power	**Class**	signal
Domain	translation	**Domain**	rotation
Position		**Position**	
Orientation		**Orientation**	
Interface	feet	**Interface**	dial
Direction	source	**Direction**	

Based on the functional specification, concepts and architectures are generated using catalog electro-mechanical elements, such as generic springs, gears, analog dials, etc. Agents in the system are coded with operations to combine the physical elements according to the demands stated in the input-output FP’s. These agents use composition rules, in concert with the physics underlying the physical elements and design domain. Other agents modify concepts as they are created, through fragmentation and iteration. In turn, still other agents manage the search process through Pareto optimization strategies to retain or discard concept paths in the MIMO design space.

Figure 9 illustrates a design (one of many) generated as a concept for the human weighing machine. Notice that the weighing machine includes the expected functional elements, at least in "stick-figure" form. It also calculates (estimates) the results of design objectives, such as cost, mass, and accuracy, which is the basis for its search strategy.

In addition to this domain, Campbell, *et al.*'s work has been applied to MEMS design. Intriguing results have been created through this application, even with a very small catalogue of physical elements to initiate the design search. This use in another domain illustrates the potential efficacy of the method.



Components:
lever: 5 cm bar stock, $1.15, w=1.0", t=0.25"
spring: ERS-A1-36, $0.93, k=16.0 lb/in
rack: KHS-F2-142, $26.75, pitch=64
gear: LAS-F7-28, $5.75, 28 teeth
bearing: ABS-A2-19, $6.99
shaft: AAS-A8-30, $2.25, d=0.25"
dial: $1.50
footpad: $1.50

Design Objectives:
cost = $46.82, **mass** = 0.2 kg
input dx = 4.1 mm, **accuracy**= 0.4 rad.

Figure 9. Example weighing machine concept (Campbell, *et al.*, 2000).

Analysis – Architecture of Campbell, et al.'s Agent-Based Synthesis Method:

Considering Figure 2, the agent-based functional synthesis method of Campbell, *et al.* shows significant coverage of the general architectural model. Their method includes a subset of customer needs, translated into FP's and objective functions. Inputs include functional specifications (again the FP's), functional elements (generic springs, etc.), and catalog structural elements (coded part numbers with associated data). The workframe for the method is the set of data structures for the methods FP's and components, Figure 9, where the geometry is not fully created. This agent-based method also includes synthesis and search strategies, as well as control, evaluation, and optimization.

Overall, the method shows a remarkable level of generative capability. It "learns" and formalizes strategies for fragmentation and component combination that are known (but only at an informal and *ad hoc* level) design principles and rules of thumb. However, most of the effort relates to the "stick-figure" concept generation. Significant functions of the designs are not represented, nor is the geometry layout fully captured. These basic capabilities will need to be explored further to satisfy the research goals. In general, the strategy seems to be inherently flexible for such extensions.

Analysis – Design Model: Again, Campbell, *et al*.'s method shows a good coverage of the design space. Physical architecture (organ structures), layout, geometry, and embodiment structures are necessary future extensions to create more competitive designs. In addition, the functional represent could be advanced. The use of the functional parameters is innovative and very powerful. However, adding product (organ) architectural and geometric issues may necessitate a more rich representation language, such as functional descriptions as opposed to mathematical specifications, exclusively.

Analysis – Research Model: The entry portal for this research is the "New Idea for a Method" path. This path shows significant creativity on the part of the researchers and could lead to significant impact on the research field. However, no descriptive efforts, such as industrial partners or industry sector studies appear to have been developed. This step, in addition to more complete research goals, is needed as the research matures further. Possible tool development will also need to be considered, especially the overhead required for user interfaces and catalogue/functional element coding. The recent extension to MEMS design shows very encouraging results along these lines.

Catalog Design: Component-Based Compositional Methods

Catalog design methods, in function-based synthesis, focus on the representation and abstraction of known structural elements to create concepts. In this section, the work of Chakrabarti and Bligh (1994, 1996a,b,c) is considered. Their approach to function-based synthesis is catalog in nature and composes designs from known components. It should be noted that the method discussed in the previous section, agent-based synthesis, could also be categorized as catalog design. In this case, however, it is categorized by its control strategy, as opposed to its inputs and representational strategy.

The component-based compositional method of Chakrabarti and Bligh represents design problems according to an input-output structure, comparatively to Campbell, *et al*. This representation is in the form of a functional specification (mathematical) as opposed to a more general functional description. It also focuses entirely on the mechanical domain, such as kinematics and mechanical power flow.

To compose a design based on the input-output specification for a machine, Chakrabarti and Bligh abstract mechanical components as input-output elements with motion transmission and motion constraints on each end. Example elements include abstracted versions of generic shafts, tie-rods, cams, levers, screws, etc. Their library of elements may be extended using their general representational approach.

The method then includes a control strategy for synthesizing the range of mechanical devices from SISO to MIMO. This strategy, referred to as "kind synthesis," exhaustively searches for chains of elements that satisfy the input-output functional specification. This computational search ends with pre-determined bounds on the number of elements or complexity of the chains. The designer is then left to interpret the chains as design concepts, embodying the concepts with geometry and interfaces to realize a physical structure. The chain concepts do, however, provide layout information in terms of orientations and positions in 3-D space.

<u>*Analysis – Architectural, Design, and Research Models*</u>: The analysis of the component-based composition method of Chakrabarti and Bligh is very similar to the agent-based approach of Campbell, *et al*. Chakrabarti and Bligh have completed some basic descriptive studies with designers, but industry data is needed, as with the agent-based method. Similar comparisons exist for the activities in the architectural model, the steps in the design process, and the elements of the research model. The main difference occurs, however, in the architectural model. Component-based composition does not include as sophisticated a control strategy as the agent-based method. Likewise, very little optimization is considered, whereas Pareto frontiers are considered in Campbell, *et al*.'s method. On the "flip-side," however, component-based composition includes more complete spatial information, with the generation of some very unique, as well as historical significant, concepts.

Set-Based Catalog Design Methods: Mechanical Compilers

Beyond component composition and agent-based synthesis, other catalog design strategies exist. The most significant is the strategy implemented in set-based methods. These methods use mathematical representations of the design problem and consider the search for solutions as an identification of the feasible subset of designs that satisfy a functional specification.

Most notable of the set-based methods is the work of Ward and Seering (1988, 1989a,b). This method takes a schematic input of a mechanical system. Besides this primary input, other inputs include mathematic representations of the functional specification for the system, as well as a catalog of elements that may be "synthesized" into the schematic to solve the specifications. The outputs from the method are sets of components that combine to satisfy the design problem.

Key features of the method are a high-level set based language for representing a systems design problem and an interval calculus for transforming the specification into lower-level structural elements. These features provide the machinery and formalism necessary to "compile" design solutions (thus the term coined as "mechanical compilers"). They also provide a control strategy for the search of design solutions, without the need for exhaustive search or unneeded enumeration of alternatives that are infeasible.

This method has been applied to relatively small-scale mechanical and sensing design problems, such as an ice cream machine, referred to as the Toscanini's Problem. The set-based method by Ward and Seering has also been extended significantly to include more powerful set-based operations, such as vector-space mathematics. It has also been extended, at least in principle, by other researchers, such as Bradley and Agogino (1994), to include uncertainty information during the catalog design synthesis.

<u>*Analysis – Architecture of the Set-Based Method of Ward and Seering*</u>: Reviewing Figure 2, Ward and Seering's set-based synthesis method begins with a mathematical functional specification (derived, at least in spirit but not explicitly, from CN's), in addition to a workframe composed of a system schematic (layout). Interval calculus and constraint propagation techniques are used to control the search for structural elements that satisfy the functional specification. No activities are devoted to abstract functional elements (besides the mathematical representation), creation of topology, or optimization and embodiment. In fact, the initial schematic plays a large role in setting the physical architecture (general topology) of the system. Alternative layouts (product architectures – organ structures) are not considered.

<u>*Analysis – Design Model*</u>: Coverage of the design process (Figure 2) by the method is focused. Because of the initial schematic input, the method concentrates on a controlled search for solutions as prescribed by the schematic. The focus is thus concept generation for a given organ structure, with little follow-up in terms of detailed topological or form (embodiment, geometry, etc.) structures.

<u>*Analysis – Research Model*</u>: The entry portal for Ward's and Seering's set-based method is the "New Idea for a Method" path. Clearly, the researchers seek an innovative approach to synthesis through compilation, a noteworthy and seminal work. However, the criteria for success are not completely defined. The method generates designs as combinations of catalogue part numbers, where feasibility is the primary criterion to note success. This criterion is met very well; yet no criteria for realizing the method in realistic scale problems are provided. Thus, industry data and studies are needed (descriptive approaches) to mature the method, realize actual designs, and identify the critical research issues.

Catalog Design Methods Using Grammars

A method that is related to that of Campbell, *et al.* (1999, 2000) is one that uses grammars to transform abstract representations of function to less abstract representations of form. This approach can be associated with the techniques developed and presented by Pahl and Beitz (1996), whereby design artifacts are represented using energy, material, and signal flows connected via a hierarchical network spanning input to output. As the design process progresses, this representation is refined from a simple black box model to a fully developed function structure (functional model) that represents all primary and ancillary functions of the artifact. A more detailed description of this representation is described in the design model section of this chapter.

In this context, grammars represent a vocabulary of valid symbols and rules used to describe the function of an artifact, using the rules to guide combination of the symbols to form meaningful expressions. The essential symbolic content of a function grammar is the representation of energy conversion relationships and the rules for combining elements that perform the conversions. The energy conversion relationships of the grammar may consist of input/output relationships such as conversion of flows: rotation→translation, human→mechanical, pressure→force, etc. The rules of the grammar will define the conditions under which these conversions may be combined to satisfy the overall functional requirement of the artifact being designed. Function grammars are currently found in association with optimally directed approaches to design synthesis (Schmidt and Cagan, 1996). These approaches use high-level optimization techniques, such as simulated annealing, within the design space search algorithm. Simulated annealing directs the search through the design space to end on the state that represents a design that is optimal, or near-optimal, in terms of quantifiable design and performance attributes.

To date, design-method work, using function grammars, has focused on machine design, the transformation of a description of the function to be performed by a machine to a description of the machine that will perform the function (Finger and Rinderle, 1989; Hoover and Rinderle, 1989; Schimdt and Cagan, 1995, 1997, 1998). Machine design is clearly a function to form application, one that seems to fit the goals of function grammar research. The following subsections highlight a representative set of works presented thus far, where each is evaluated against the architecture, design, and research models developed above.

Abstraction Grammar Methods

The work of Schmidt and Cagan (1995) highlights the abstraction grammar, the first function grammar to be discussed. This work is a predecessor of the A-Design (agent-based) synthesis method discussed above. In Schmidt and Cagan’s approach, abstraction grammars are developed as a basis for the larger framework of the Function to Form REcursive Annealing Design Algorithm (FFREADA), which is an optimally directed search approach to design synthesis. Within this technique, abstraction grammars are defined as “a production system for the representation and generation of function and form layouts.” The abstraction grammar representation system uses a library of function and form entities from which selections are made to satisfy input/output specifications at the current level of abstraction. Several suggestions are made for the system of representation, such as bond graphs (Paynter, 1961) or the energy, material, signal approach of Pahl and Beitz (1996), but none are prescribed, the authors preferring to allow the user to select a representation system appropriate to the domain of the problem. The representation system will manifest itself in the library of function, form, and rule entities developed for the application at hand.

In general, the function and form of the artifact are initially represented at a very high level of abstraction; the authors use the energy – material – signal representation of Pahl & Beitz (1996) to represent a black box model in their examples. Initial specifications are in the form of target values for the output function. In this example, system cost is to be minimized and the output torque is to be at least 1.36 N-m. The functional specifications are given, so the left side of the arrow is the input, and the right side is the output. The drill power train example is thus specified as shown in Table 7. This initial specification is built upon and refined as the synthesis process progresses.

Table 7. Initial specification for FFREADA (Schmidt and Cagan, 1995).

CONSTRAINTS	**ENERGY**	**MATERIAL**	**SIGNAL**
Torque >= 1.36 N-m	0→R	0→1	1→0
Minimize Cost	0→C		

This specification indicates energy input is zero, and the output is Rotation that is Continuous. Zero material (“0”) is input, and a drill bit (“1”) is output. A trigger signal is input as indicated by the 1, and no signal is output.

The rules of the grammar are applied to ensure that compatibility between form and function entities is maintained. Abstraction grammars are string grammars; relationships contained within the grammar are sequential in nature. The rules then must ensure that each element in a string is compatible with the previous and subsequent elements so that generated strings represent valid machine elements or assemblies. Having satisfied the rules at one level of abstraction, the abstraction grammar approach proceeds to the next lower level of abstraction, considering Figure 10. The essence of moving to a lower level of abstraction is that the form and function entities contain more detailed information. Designs at the next lower level of abstraction are instantiations of the previous level, with several designs being possible valid instantiations for the higher level. This process is repeated through *m* levels of abstraction until level-0 is reached. Level-0 represents a component topology and is the final design or set of designs generated by FFREADA.

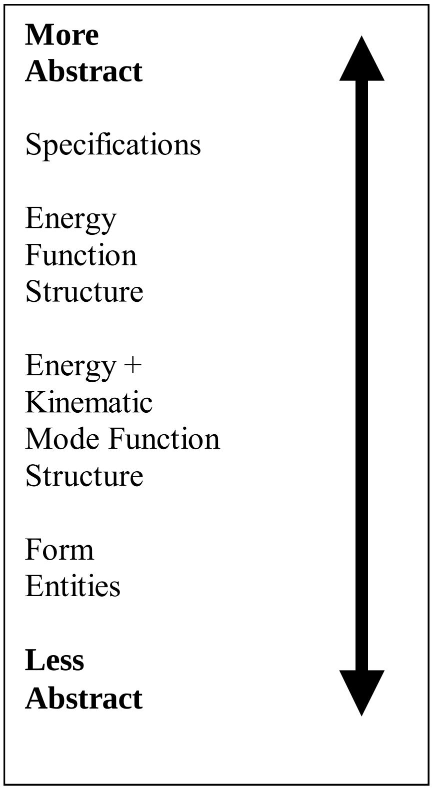


Figure 10. FFREADA Design Model (Schmidt and Cagan, 1995).

The authors highlight several observations regarding the approach. The sequential nature of string grammars limits the types of machines that can be represented using this approach. Sequential machines are still an acceptable way of representing an artifact such as a transmission, but there are clearly limitations. The rule set results in designs biased toward minimizing the number of components in a machine. Finally, the designs are functionally segmented, i.e., each function is discretely instantiated by a single machine component. This observation leads to two further consequences:

- Synthesis of explicit function sharing is not possible.
- Mapping of performance metrics of specific components up through the function structure, to the functions they satisfy, is possible.

<u>*Analysis – Architecture of Abstraction Grammar Methods*</u>: Considering Figure 2, this method provides an interesting synthesis strategy to abstract elements. However, in many respects, it is incomplete. An overview of the architecture model of Figure 2 reveals that customer needs are not addressed explicitly by the method, but they are implied in the functional specification. The method promotes the use of functional specifications as the initial description of the artifact.

Abstract functional elements are represented by the high level function and form elements, and an optimally directed search is performed on the library of primitives in order to synthesize the abstract structure. At each level of abstraction, the grammar rules are used to ensure the input/output relationships are satisfied. The method represents structural elements at abstraction level-0 and uses them in an optimally directed search of the design space to synthesize variants on the component topology. At each level of abstraction, application of the grammar rules ensures the input/output relationships are satisfied. In addition, the embodiment is optimized using the rules of the grammar. These rules do not address optimization of the topology, but do address minimization of the number of elements used to satisfy the specifications.

Analysis – Design Model: A black box description is used as the starting point of the design process in this method. In this method, the function structure is limited to series configurations based on the nature of the string grammar used to describe function and form. As a result, the functional system-architecture is SISO. The algorithm used does not have sufficient detail to develop an organ structure, hence issues such as function sharing are beyond the capabilities of FFREADA. Function sharing is one of the identified shortcomings of the approach. A topological component structure is treated in a rudimentary way as the result of abstraction level-0, component level design. There are no specific geometric details that result from this abstraction level, only hierarchical relationships between the components. Finally, the form system architecture is not developed, the authors having chosen to limit the scope of the project to abstract and structural elements. Overall, the FFREADA method represents a high-level conceptual design approach that produces multiple concepts without significant detail about instantiation.

Analysis – Research Model: This method enters the model through the New Idea for a Method model portal. The method is "a computational model for optimally directed conceptual design of machines in which the transformation of function to form occurs iteratively along an abstraction continuum on a number of defined levels of abstraction" (Schmidt and Cagan, 1995). The development approach taken by the authors is limited primarily to the mechanical engineering and mathematical/statistical domains. Tangential reference to the cognitive processes of designers during the act of synthesis is made, but no references are made to research in that area. Developed of the method is based on an assumed need for computationally-based conceptual design in industry, but does not elucidate the foundation for this assumption. There are no references to descriptive observations or analysis of industry practices and needs. An academic tool is developed, FREADA, to support the method, and this tool is tested against a limited set of success criteria through an example design of a power supply. In addition, the search and simulated annealing aspects of the tool are exercised to determine the benefit of these techniques over a purely random search of the design space. The results indicate the optimally directed nature of the tool allows it to produce optimum solutions in less iteration than is required to achieve the same results using a purely random exploration of the design space. Since the method does not take the design process to embodiment, no attempt is made to observe and analyze the results in an industrial setting, thus no feedback mechanism is generated for descriptive analysis of the method.

Abstraction Grammar Methods II: Extension of FFREADA

Further evolution of this method is carried out in Schmidt and Cagan (1998). The unique contribution of this work is the application of the technique to a design problem using drill power trains as an example. The drill power train is a sequential machine type design problem, one that the FFREADA method is well suited for. The results are encouraging. The method produces results that are consistent with commercially available products from Black & Decker® and DeWalt®. In addition, a specific catalog of machine components is specified and searched during the design process. The catalog is limited to energy, material, and signal entities and the costs associated with the components that realize each of the functions. The method results in designs that satisfy the functional specifications, but doesn't go any further than the original work toward defining a specific topology in the final design. This work is a move toward realizing a descriptive test of the method against industry based success criteria, with the logical next step being use of the method in a live product design effort.

In summary, the work done in developing FFREADA leads to some significant results. The first result is that an algorithmic method can generate feasible serial designs given a library of components that represent the domain of the desired design, drill motors and their drivetrains in this case. These designs are optimal or near-optimal in configuration as compared to products available on the market today.

Graph Grammars

In response to the limitations imposed by the use of string grammars in the two previous works, Schmidt and Cagan (1997) developed a method for treating a basic level of function sharing in the design artifacts that result from application of the method. In this method, the string grammar is replaced by a graph grammar.

The method is called Graph Grammar REcursive Annealing Design Algorithm. According to the authors, "GGREADA uses a graph-based abstraction grammar to create designs and a recursive simulated annealing process to select a near-optimum design from those created" (Schmidt and Cagan, 1997). A graph grammar is a mathematical formalism for representing graph vertices and edges. The graph grammar representation allows non-serial component arrangements and functional relationships, thus including the possibility of function sharing in the resulting designs. The overall approach to design is similar between the string grammar and graph grammar implementations, with the grammar consisting of both form and function entities and a set of rules. The design evolves from an abstract black box model to a component level topology through various levels of abstraction, each level being an instantiation of the previous level (refer to Figure 10 of the previous method). The rules of the grammar ensure the final design satisfies the initial functional specifications.

At this point in its development, GGREADA's rules do not include any type of geometric reasoning. The implication of this omission is that the resulting design satisfies the functional requirements, but still requires interpretation from the designer to ensure the design physically provides the desired behavior. Some uncertainty about the behavior is reduced by the use of a library of components. This library includes geometric content, resulting in a limited number of possible component combinations. Additional reduction of uncertainty can be achieved through the implementation of more sophisticated rules for geometric reasoning, but designer interpretation will still be required.

Table 8. Function specifications for GGREADA, Meccano® Erector® Set Carts.

FUNCTION SPECIFICATIONS		FUNCTION APPLICABILITY
1	Create Rolling	N/A
2	Support Load	N/A
3	Mount 2 parallel wheels	1
4	Mount 1 wheel	1
5	Provide surface area	2

As an example of the power of GGREADA, the method is applied to the design of Meccano® Erector® Set Carts. A sample of the required input specification is shown in Table 8. Functional applicability refers to the primary functions supported by the sub-functions in rows 3, 4, and 5. Using a limited set of components (12), an example is undertaken to design a cart given the functional requirement to “create rolling.” Given this simple functional requirement, the combinatorial nature of the catalog design approach becomes quite obvious, as over 130,000 designs are possible with the design space growing as more components are included in the catalog. This issue highlights the need for a technique to reduce the size of the design space that is searched. Simulated annealing is the technique chosen for GGREADA. Simulated annealing directs the search through the design space leading to an end state representing optimal or near-optimal results based on quantifiable performance attributes.

Analysis – Architecture of Graph Grammar Methods: Again, customer needs are not explicitly addressed in this work; they are implicitly represented in the functional specifications used as the initial description of the artifact for the method. Abstract functional elements are represented by the high level function and form elements, and an optimally directed search is performed on the library of primitives in order to synthesize the abstract structure. At each level of abstraction, the grammar rules are used to ensure the input/output relationships are satisfied. As with FFREADA, GGREADA represents structural elements at abstraction level-0 and uses them in an optimally directed search of the design space to synthesize variants on the component topology. The embodiment is optimized using the rules of the grammar, but these rules do not address optimization of the topology; in fact a multitude of topological configurations are generated and left to the designer to sort out.

Analysis – Design Model: The process description is not explicitly addressed in this method, rather it is assumed to have been established prior to initiating the design effort at the conceptual level. Based on that assumption, the black box description is used as the starting point of the design process in this method. Functional models (function structures) within GGREADA are multipath in nature, allowing function sharing and MIMO systems to be synthesized. A topological component structure is generated as a result of the geometric data contained in the component library and is the output at abstraction level-0, component level design. A basic level of form system architecture results from the geometric data. Designing using Meccano® Erector® Set Carts in the example made the resultant architecture especially apparent.

Analysis – Research Model: Similarly to FFREADA, this method enters the model through the New Idea for a Method portal. The method is "a computational model for optimally directed conceptual design of machines in which the transformation of function to form occurs iteratively along an abstraction continuum on a number of defined levels of abstraction." The development approach taken by the authors is limited primarily to the mechanical engineering and mathematical/statistical domains. The method is developed based on the limited success of FFREADA, and seeks to extend the work toward a tool that can be validated in industry. There are currently no references to descriptive observations or analysis of industry practices, but the work using the Meccano® Erector® Set Carts is a step toward validation and the establishment of a feedback mechanism for descriptive analysis of the method.

In summary, GGREADA overcomes, at least in part, the inability of a string grammar to deal with function sharing. A comparison of the two methods is difficult because the examples presented are very different in function and form. An interesting comparison might be to use the GGREADA method to design a serial device such as a drill motor, and compare the results. The method and example clearly demonstrate the feasibility of the graph grammar design process and the efficacy of the simulated annealing process, albeit on a limited scope.

Supplementary Function-Based Synthesis Methods

The analysis of function-based synthesis methods, in this chapter, focuses on the areas of dynamic systems synthesis, agent-based, and catalogue design. Specific methods and research are considered in each of these areas. These particular methods are chosen based on their seminal (original) nature or their unique contribution to one or more of the activities in the method architecture (Figure 2).

Other methods and contributions exist in this field, besides those analyzed above. Such methods and contributions include the works of Carlson-Skalak, *et al*. (1998), Finger and

Rinderle (1989a,b), Frecker, *et al*. (1997), Hoover and Rinderle (1989), Kannapan and Marshek (1987, 1991), Kota and Chiou (1992), Navinchandra, *et al*. (1991), Welch and Dixon (1992, 1994), and Schmidt, Shetty, and Chase (1998). These works fall within the areas of function-based synthesis, as defined in Figure 4. They also follow, generally, the model analysis results of the particular methods reviewed above.

IV. BASS: TECHNICAL DESCRIPTIONS OF REPRESENTATIVE FUNCTION-BASED METHODS

The previous section develops "scaffolding" for understanding the body of research in function-based synthesis. Significant data exists in this scaffolding; however, before making inferences from this data, the technical depth of the field should be investigated further. Representative methods from the field (Figure 4) will aid us in this investigation.

Two methods are presented below. The choice of these methods is based on a number of factors: (1) the most recent methods typically build upon principles learned in the original research; they are thus the most sophisticated; (2) while presenting depth, the breadth of the field (Figure 4) must not be sacrificed; and (3) it is important to understand the range of activities in the architectural model (Figure 2). The impedance synthesis method by Connolly and Longoria (2000), and A-Design by Campbell, Cagan, and Kotovsky (1999, 2000) address these factors.

Dynamic Systems Synthesis: Connolly's and Longoria's Impedance Synthesis Method

The impedance synthesis method of Connolly and Longoria seeks to advance classical synthesis techniques for electrical network theory (Foster, 1924; Brune, 1931; Baher, 1984), focusing on systems with active elements. Active elements, in contrast to passive elements, provide power flow into a system. Such elements are typically realized by actuators, in conjunction with appropriate physical control of the power source. The synthesis of such elements represents a significant advancement of dynamic systems synthesis methods. The inclusion of active elements can potentially simplify the complexity of devices, reduce weight and cost, and increase the level of automation (active control).

The driving impetus of the method is to synthesize a collection of resistive, inductive, capacitive, transformer, gyrator, and power source functional elements to satisfy a desired dynamic response of a system. This synthesis must follow physical power laws, and is applicable to multiple energy domains. After synthesizing the functional elements, the method then seeks a physical realization of these elements. The goal is to create alternative networks of functional elements to satisfy the desired response. Physical simulation of the system drives the creation and selection of these networks.

For context, let's consider Connolly's and Longoria's method with respect to the architectural model in Figure 2, building on the summary and analysis in Section III. Their method focuses on a subset of the activities in the architecture: functional specification, creation of a workframe, synthesis of functional-element structures, evaluation and search, and control of the synthesis. Functional specifications are in the form of a desired frequency response, stated as an impedance function. The workframe is an abstract functional space, i.e., a power-flow network in terms of bond graph elements. The initial workframe commences with an initial bond graph of the partial passive system. Synthesis of these elements occurs through mathematical manipulation of the input functional specification to create an equivalent bond-graph system (functional-element structure). Alternatives are generated through directed replacement of the functional-elements with combinations of passive, active, and control elements. These alternatives are created, selected, and controlled based on simulation of the bond-graph representations and physical rules for replacement.

After synthesizing physically realizable functional designs, a "catalogue" approach is taken to replace the functional elements with structural elements. Physical components replace the functional elements in the bond graph. This process is directed and controlled by the designer, where no formal representation of the structural elements is provided. The result of this process is a "skeletal" schematic of the structural system. No embodiment is carried out as part of the method.

Procedure/Algorithm

Based on this architectural summary, Figure 11 presents the organization of Connolly's and Longoria's synthesis method. The following sections detail the steps shown in Figure 11. A motivating design example, i.e., an active vehicle suspension, is used to illustrate the application and results generated for each step (Connolly and Longoria, 2000).

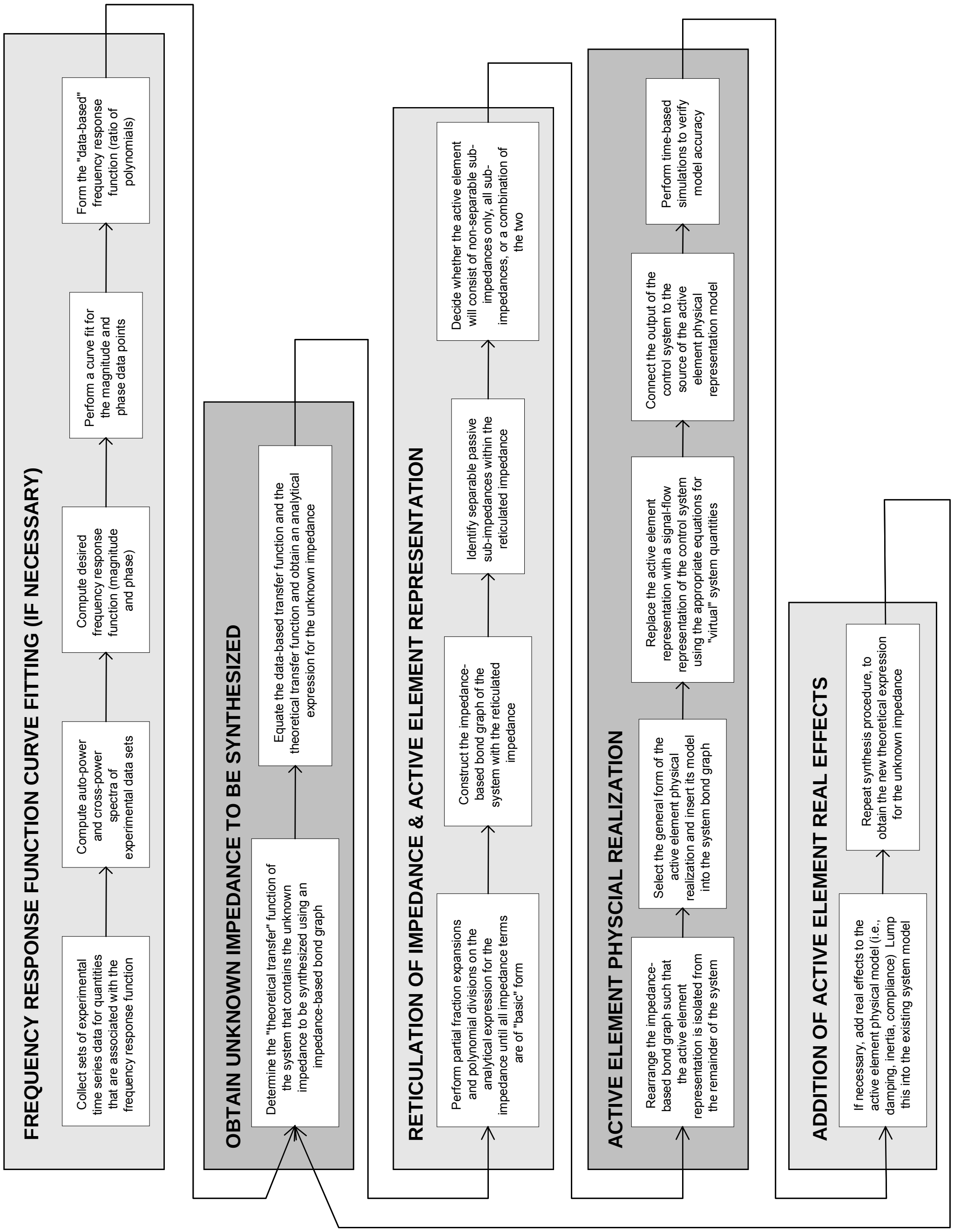


Figure 11. Active element synthesis procedure flowchart (Connolly, 2000).

STEP 1 - Create Input Functional Specification:

The first step involves developing a functional specification for the design. In this case, the functional specification should be in the form of a frequency response function: magnitude (dB) and phase (deg.) versus frequency (Figure 11). This response function may be prescribed as an ideal relationship from the customer needs, or, alternatively, it may be constructed empirically through experiments and curve fitting.

Application: To demonstrate the method, the system under study is a quarter vehicle configuration that consists of a tire, an unsprung mass and a sprung mass, between which there is an original electromechanical suspension system. This suspension system consists of an air spring and an electric motor, coupled with a rack and pinion gearing system controlling the position of the sprung mass. Figure 12 shows a double exposure photograph of an experimental setup for this system.

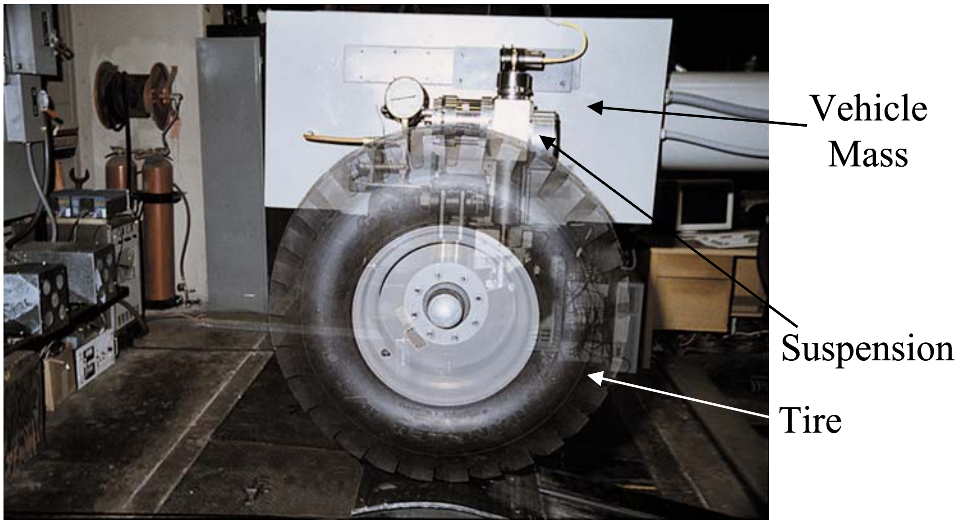


Figure 12. Photograph of quarter vehicle experimental setup (Connolly, 2000).

The parameters of the system are: sprung mass, m_{sm} = 613 *kg*; unsprung mass, m_{um} = 81 *kg*; suspension spring stiffness, k_p = 35,550 *N/m*; effective tire stiffness, k_t = 250,000 *N/m*; and effective tire damping coefficient, b_t = 4000 *N-s/m*. Experiments are run with this system to plot a frequency response (not shown here). It is desired to replace the existing electromechanical actuator with an actuator that involves another combination of energy domains, such as hydraulic-mechanical. Thus, the active-element synthesis method of Connolly and Longoria is applied, in this case, to an adaptive redesign problem. Figure 13 illustrates a schematic of the synthesis problem with an unknown suspension system.

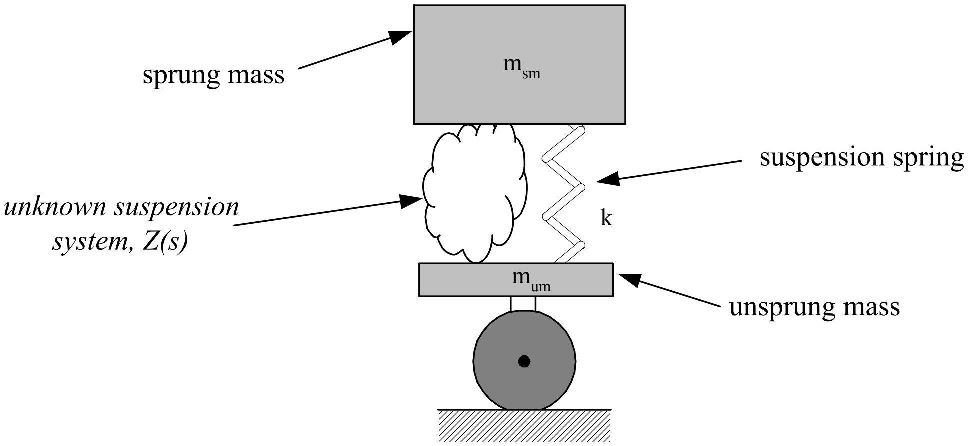


Figure 13. Actuator replaced by an unknown suspension system (Connolly, 2000).

STEP 2 - Transform Functional Specification to an Unknown Impedance Representation:

The next step of the method involves casting the desired frequency response into an unknown configuration with an impedance given by *Z(s)*. This process involves representing both the desired frequency response and the unknown system (Figure 13) as transfer functions. Equating the transfer functions leads to a mathematical expression for the impedance.

After specifying the *Z(s)*, either empirically or as prescribed, reticulation is applied to the impedance to create its basic terms. The process of reticulation involves the application of decomposition and partial-fraction expansions to the impedance function, followed by the synthesis of bond-graph elements that are equivalent to each basic term in the decomposed *Z(s)*. The output for Step 2 is only the first-stage of the reticulation, i.e., the decomposed impedance function.

Application: Impedance decomposition for the suspension problem yields the impedance function in Eqn. 3, whose terms consist of basic impedances only. Notice in the equation that the superimposed basic terms include both negative and positive real values. The negative terms correspond to active elements.

$$Z(s) = 9072.4 - \frac{35550}{s} + \frac{1}{-0.00011 - 6.08178\times10^{-7}s} + \frac{1}{-0.03122 - 0.00960s + \frac{1}{2.64997 - 0.93526s}} \tag{3}$$

STEP 3 – Synthesize Active-Functional Element Representation:

The second stage of reticulation is applied to the decomposed impedance function to synthesize a functional-element representation of the desired dynamic system. The elements in the representation include both passive and active elements, depending on the basic terms in *Z(s)*.

This synthesis procedure involves the one-to-one mapping of the basic terms to energy-domain-independent resistors, capacitors, etc. These elements are connected with power bonds following the procedures of bond-graph modeling.

Application: Figure 14 shows the bond graph structure that represents the synthesis of the suspension system, equivalent to Equation 3. Active elements in this representation are illustrated as bond-graph elements with prefaced negative signs.

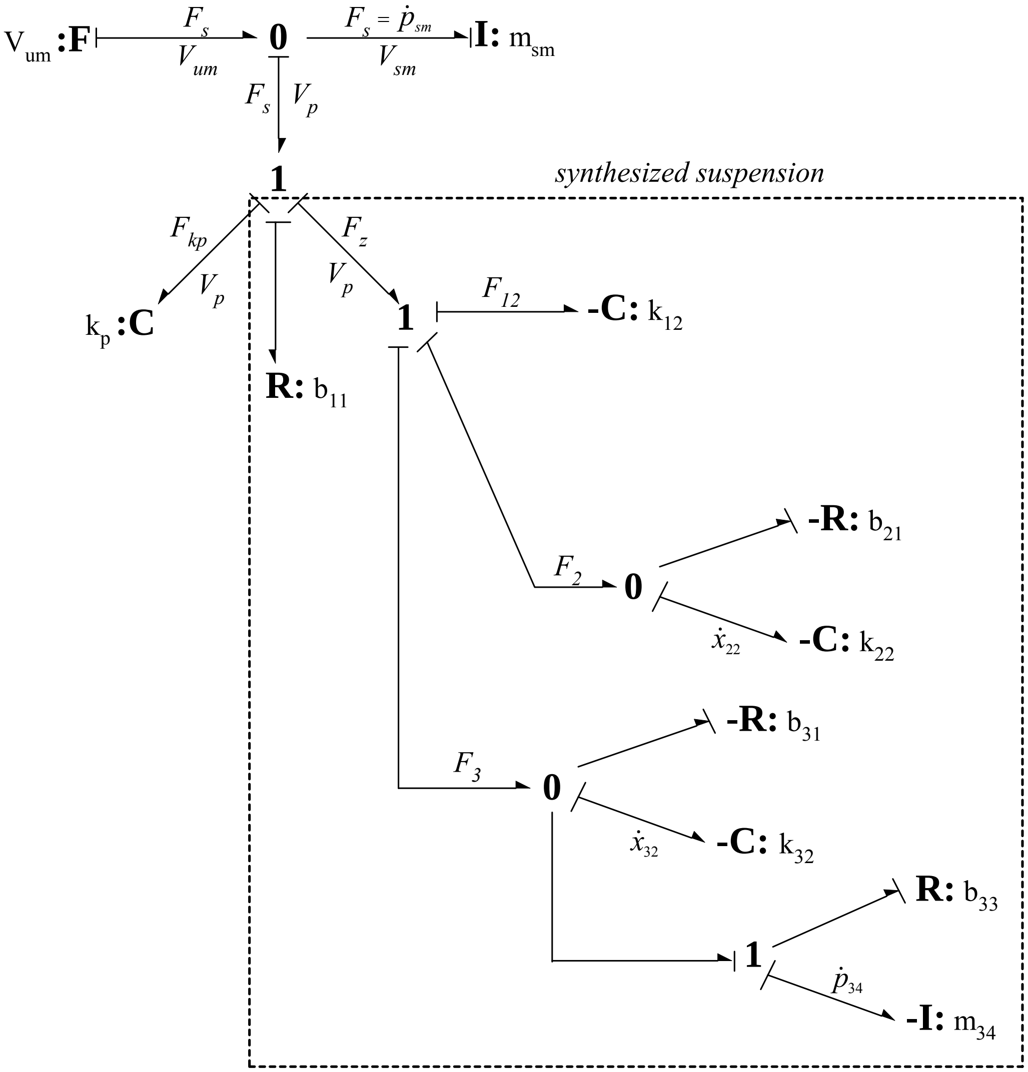


Figure 14. Bond graph representation of synthesized suspension system (Connolly, 2000).

<u>*STEP 4 – Synthesize Active-Functional-Element-Physical Realizations (Including Control)*</u>: The second stage of synthesis is now executed, i.e., the composition of functional elements that are physically realizable. This step involves a number of sub-steps. First, separable passive elements in the synthesized active system (if they exist) are either retained in the active system or

rearranged and combined into the remainder of the system (i.e., the non-active subsystem). This approach produces alternative concepts for the design problem. Next, physical bond graph representations are used to replace the negative bond graph elements (which are, without replacement, not physically realizable). These physical bond graphs represent physical devices (such as motors and gear trains, hydraulic actuators, etc.), all of which require separate power input. Replacing the negative bond graph elements with different actuator representations results in alternative concepts, depending on the number of actuators that are catalogued with physical representations.

After replacing the negative elements, control elements are synthesized for the alternative active systems to produce, theoretically, the desired output. These control elements, in block diagram form, regulate the behavior of the active elements as they interact with the remainder of the system. The control subsystem performs this regulation through signal flows.

As a final sub-step, physical simulations are carried out on the synthesized bond-graph configurations to select preferred configurations and to detect performance that deviates from the desired output. Rules can be applied, based on the simulation results, to add “real” effects to the active-system models, and improve the expected performance of the designs. With the addition of real effects, the synthesis process iterates to create a new theoretical impedance of the synthesized alternatives. Steps 3 and 4 are then repeated to converge to preferred designs.

Application: For the suspension design, Figure 14 shows the synthesized active system, retaining a passive element R: b_{11}. Alternatively, Figure 15 shows a different concept, where the separable passive element is rearranged out of the active subsystem, creating a hybrid configuration.

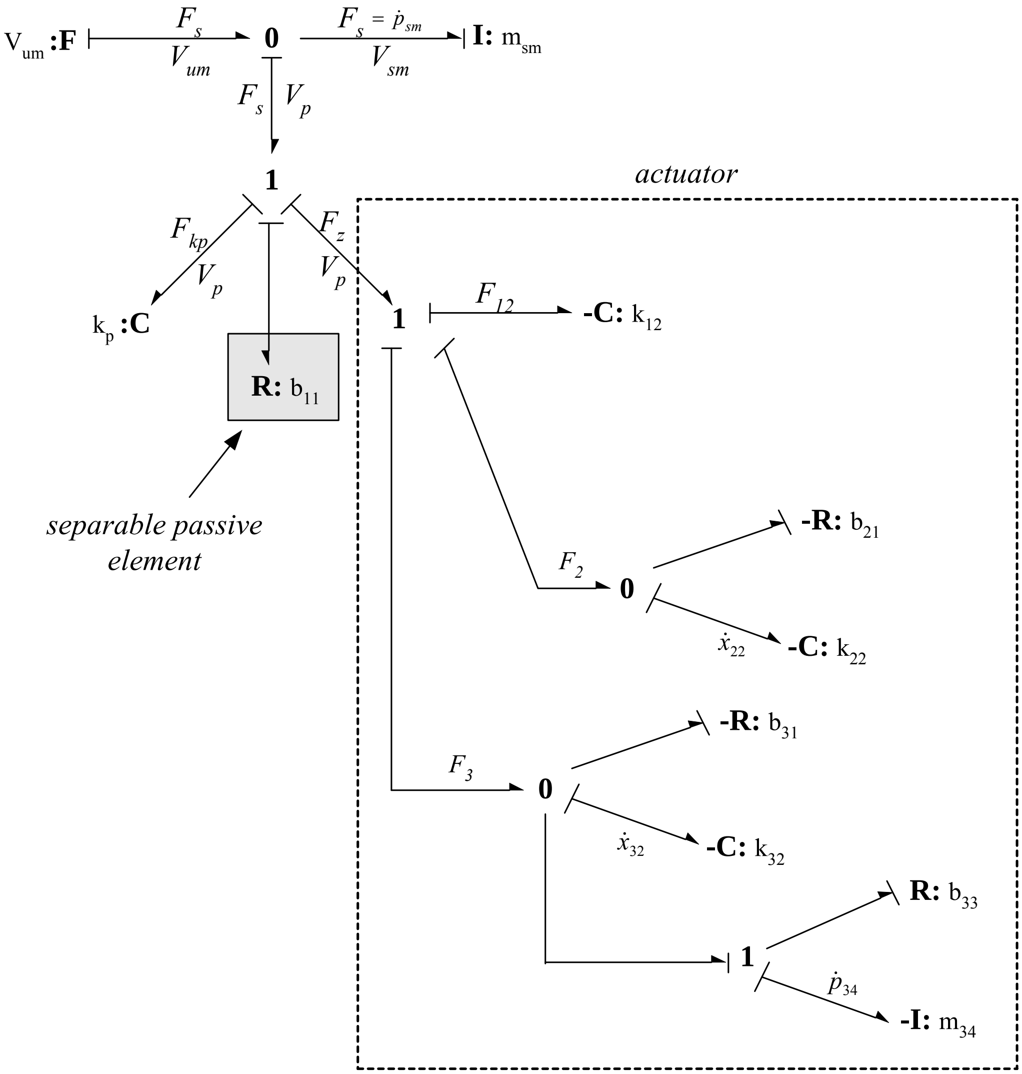
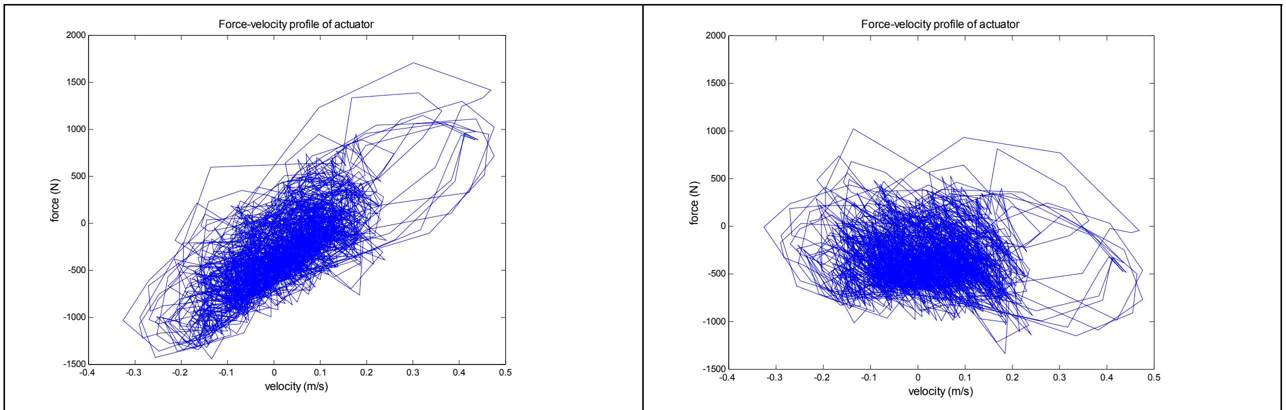


Figure 15. Hybrid suspension actuator representation (Connolly, 2000).

We may simulate these alternative configurations, using parameters for the elements, as derived from the impedance function. The resulting force-velocity profiles due to random excitation for the fully active and hybrid cases are shown in Figure 16. The profile for the hybrid case is notably flattened, indicating that the power requirements for the actuator in this case are comparatively lower. Alternatively, the range of the actuator force for the fully active case is much wider than that of the hybrid case. In many cases, an actuator requiring lower power is more desirable. However it is important to investigate both cases, since there may be a situation in which a design constraint, such as available space, may rule out a hybrid system.



(a) Fully-active actuator (Figure 14).

(b) Hybrid actuator (Figure 15).

Figure 16. Force-velocity profile for fully active and hybrid suspension actuators (Connolly, 2000).

We now synthesize the physical realizations of the negative bond graph elements. This step entails replacing the negative elements with physical-actuator models. We also generate the control elements for the actuator. Considering the fully active configuration, for example, and choosing a hydraulic actuator as the active element, the synthesized bond graph of the system, along with signal flow for the controlled effort source of the actuator, is shown in Figure 17.

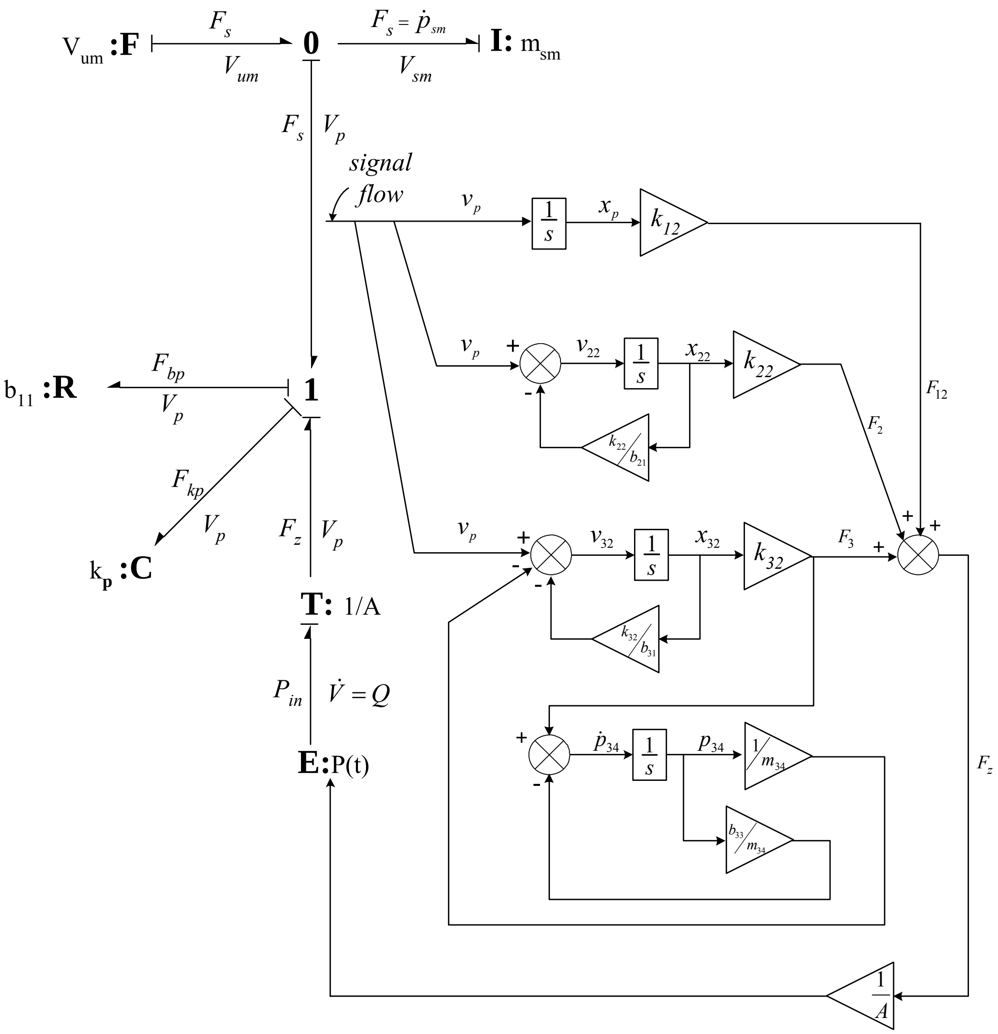


Figure 17. System bond graph with signal flows and controlled effort source (Connolly, 2000).

For the concept shown in Figure 17, real effects may now be added. Figure 18 illustrates the addition of real effects associated with the actuator piston, such as piston mass and viscous damping, to the bond graph. A piston mass, $m_p=1\ kg$, and a piston viscous damping coefficient, $b_p=100\ N\text{-}s/m$, are selected.

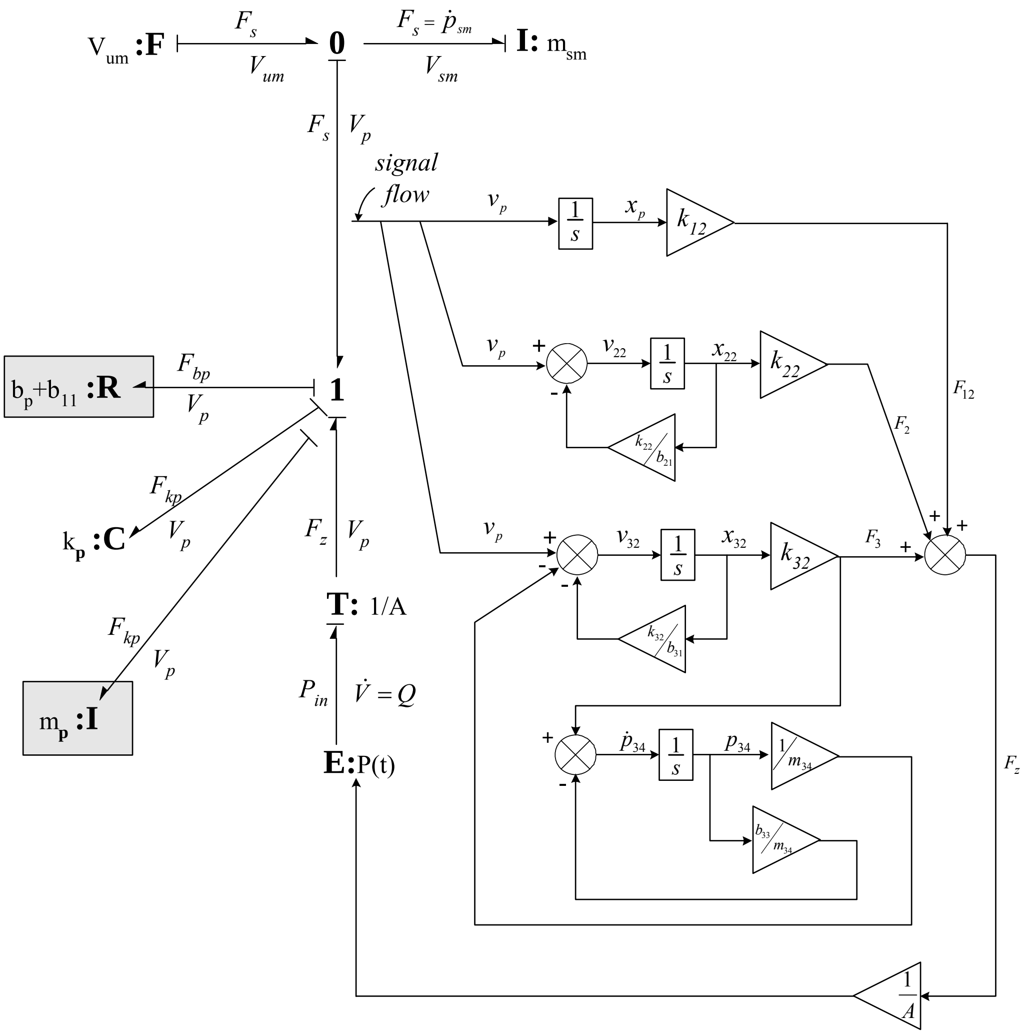


Figure 18. Bond graph illustrating real actuator effects (Connolly, 2000).

STEP 5 – Synthesize Structural Elements (manual):

The final step in Connolly's and Longoria's method, as least conceptually, is to synthesize structural elements to replace the bond-graph representation, as in Figure 18. Figure 19 shows a skeletal schematic of the hybrid design configuration. This design must now be embodied (not covered as part of the method).

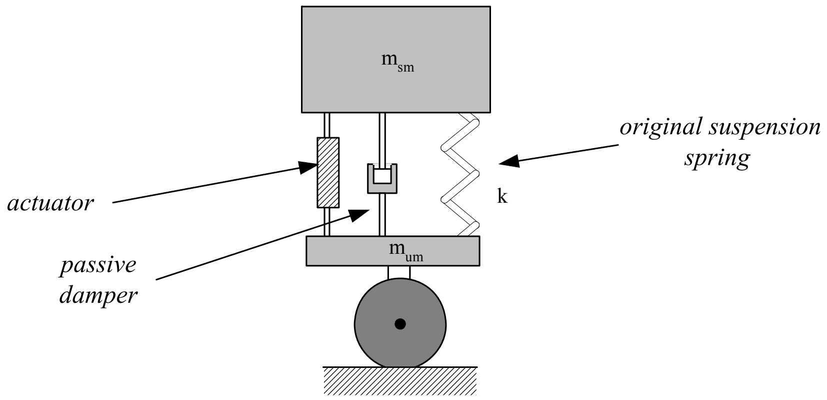


Figure 19. Hybrid configuration for synthesized suspension system (Connolly, 2000).

Campbell's, Cagan's, and Kotovsky's Agent-Based Synthesis Method

The active-element synthesis method of Connolly and Longoria focuses on the algorithmic synthesis of dynamic systems. They use and extend the bond-graph language to create alternative design concepts. However, their level of automation, control, and optimization is fairly rudimentary. In this section, a representative synthesis method is detailed that focuses on these complimentary activities (Figure 2). The chosen method for this study is the agent-based method (A-Design) of Campbell, Cagan, and Kotovsky (1999, 2000).

Building on the summary in Section III, the driving impetus of Campbell, *et al.*'s method is to generate design configurations through software agents. Their approach contributes significantly to the general field of computational synthesis, where their techniques are, fundamentally, domain independent.

Procedure/Algorithm

Figure 20(a) illustrates the general approach of Campbell, *et al.*'s agent-based method. Figure 20(b), in turn, refines this approach for the application domain of electromechanical systems. Each step of the method is detailed below, focusing on the implementation of A-Design in Figure 20(b). While the steps of this process are presented sequentially, many of the activities in Figure 20 can be carried out in parallel.

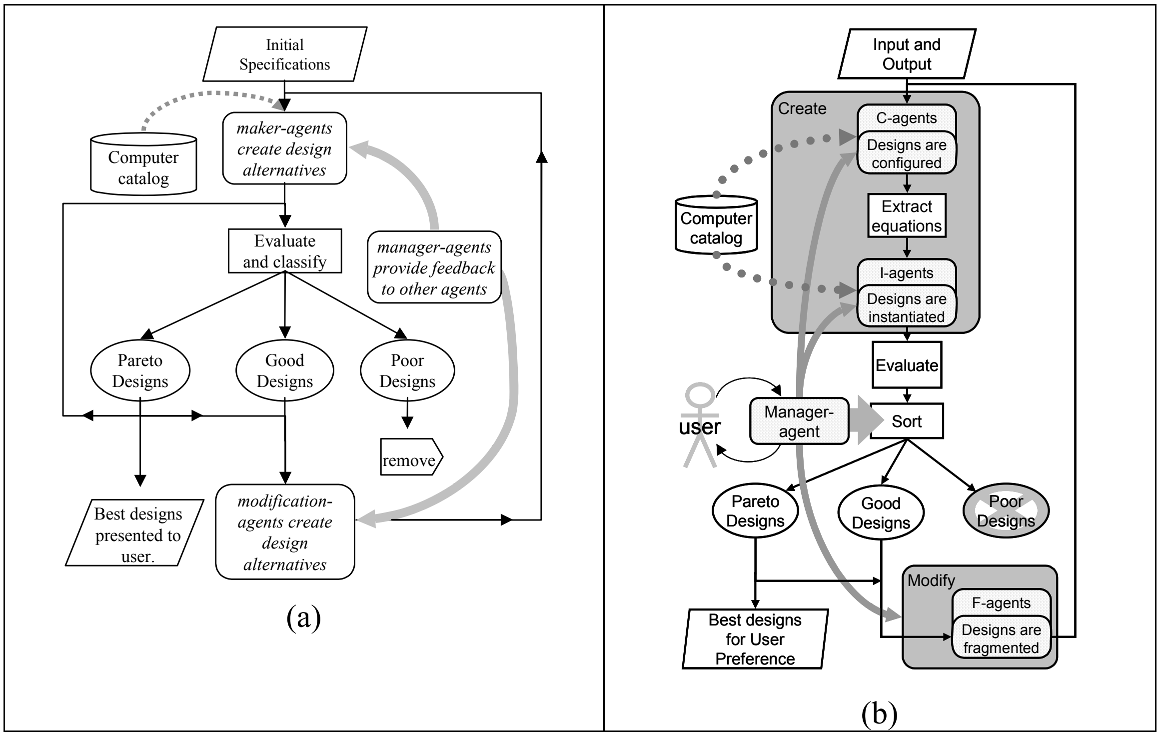


Figure 20. Agent-based synthesis method: flowcharts (Campbell, *et al*, 1999).

STEP 1 - Define Input Functional Specification:

Initially, the designer or design team defines a functional specification. This specification is in the form of input and output functional-parameter (FP) structures. An FP contains variables that describe the desired electro-mechanical behavior. In Campbell, *et al.*'s representation, components are described by ports, or points of connectivity, that define the interaction with other components. FP's are then the interface, i.e., input-output, representation of electro-mechanical devices. These interfaces contain the constraints, energy flow, and signal flow information describing the components. Table 5 presents the contents of a generic FP structure. Table 9, in turn, lists the specific relationships for the *through* and *across* variables of an FP for different energy domains.

Table 9. Definitions of FP *through* and *across* variables for different energy domains (Campbell, et al., 2000).

	Translational	Electrical	Rotational	Hydraulic
Through Variable	Force (f [Newtons])	Current (I [amps])	Torque (T [N-m])	Flow Rate (m [kg/s])
Across Variable	Velocity (v [m/s])	Voltage (v [volts])	Angular Speed (Ω [rad/s]_	Pressure (P [Pa])
Through $\propto$ Across	Damper friction	Resistor	Damper friction	Valve viscous drag
Through $\propto$ d(Across)/dt	Mass	Capacitor	Rotational Inertia	Tank
Through $\propto$ $\int$ (Across) dt	Spring	Inductor Coil	Rotational Spring	Long Piping

Application: To illustrate the agent-based synthesis method, the design of a machine to weigh humans is considered. Figure 21 and Table 6 define the input specification, according to the FP formal representation. The input is a downward force, supplied by a specimen's weight, and the desired output is an angular displacement of a dial indicator.

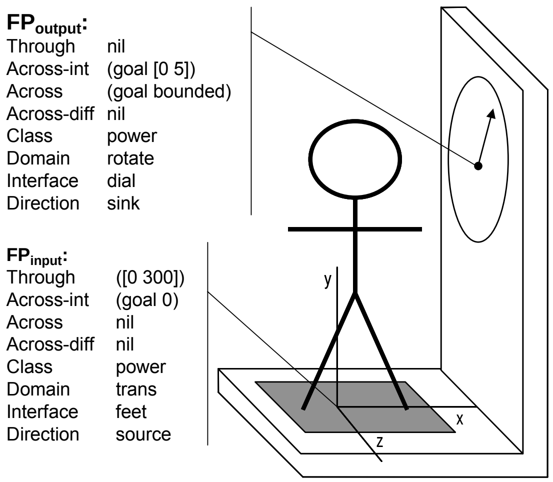


Figure 21. Weighing machine design problem – Functional Specification (Campbell, *et al.*, 2000).

<u>*STEP 2 – Configure (Synthesize) Functional Designs with C-Agents*</u>:

After defining the functional specification, the configuration agents (Figure 20(b)) synthesize design configurations by adding one component (represented with FP's) at a time to a given configuration until the desired input-output specification is functionally satisfied. C-Agents compose the design configurations through a reasoning scheme, where they add a component embodiment, either in series or parallel, to existing components at a given state of the configuration development. The goal is to add components that fully satisfy the input-output specification. If this is not possible, components are added to input-output ports that further the design toward its requirements. A combination of decomposition (problem simplification), goal-oriented, and random-search strategies are adopted by the agents to synthesize the design configurations.

An important input into this synthesis process is a catalogue of components (referred to as embodiments – EB's – by Campbell, *et al.*), from which the C-Agents create configurations. This catalogue contains functional representations (FP's) of physical components, such as gears, springs, levers, and dials. A given catalogue depends on the domain of application, but covers, in general, multiple energy domains and atomic (one-function) behavior. Table 10 lists the current components (embodiments) implemented in the A-Design testbed.

Table 10. Current components represented in the A-Design testbed for electro-mechanical systems (Campbell, *et al.*, 1999, 2000).

battery	cable	capacitor	electrical valve
spur gear	inductor coil	lever (class 1)	lever (class 2)
lever (class 3)	motor	pipe	piston
potentiometer	pulley	rack	relay
resistor	rotational bearing	rotational damper	rotational valve
shaft	solenoid	spring	sprocket
stopper	switch	tank	torsional spring
transistor	translational bearing	translational damper	worm gear

Application: Figure 22 illustrates the synthesis of a weighing machine configuration using C-Agents. A spring component is added to the configuration to make it complete, satisfying the input-output specification (Figure 21).

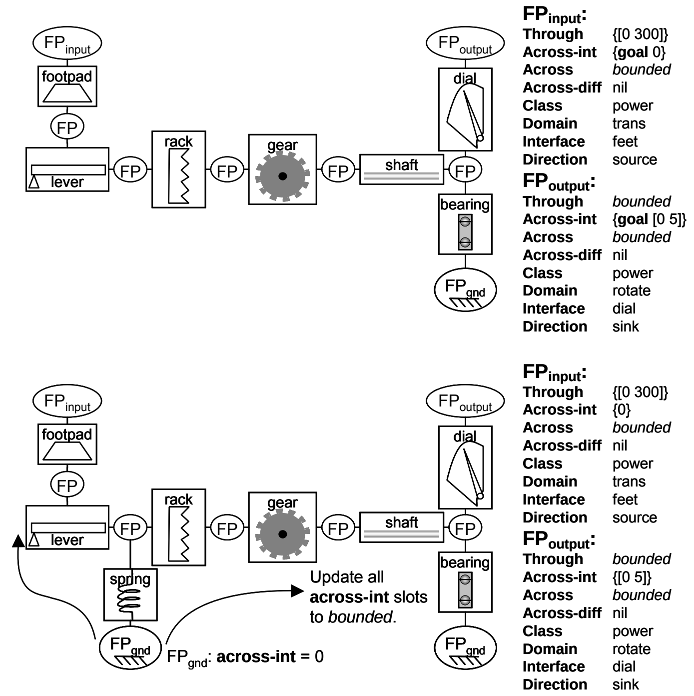


Figure 22. Synthesized configuration: (a) partial design state with FP's; (b) completed design with the addition of a spring component (Campbell, *et al*., 2000).

STEP 3 – Extract Behavioral Equations:

Once a design configuration is completed, behavioral equations are extracted, in symbolic form, for the goal states of the system. Equation extraction is completed by recursively computing a system state by starting at a goal state and working backward to given data at ground FP's (Campbell, *et al*., 1998). The extracted equations are used to evaluate how well a configuration satisfies the design specification. The agent-based method uses these equations, and their evaluated values, to synthesize, iteratively, updated and improved configurations.

Application: The extracted equations for a weighing machine configuration are given by the goal dial displacement and goal input displacement, as shown in Equations 4 and 5.

$$\theta_{dial} = \frac{1}{r_{gear} k_{spring}} \frac{d_1}{(d_1 + d_2)} F_{weight}, \tag{4}$$

$$x_{input} = \frac{1}{k_{spring}} \frac{d_1^2}{(d_1 + d_2)^2} F_{weight}, \tag{5}$$

where θ_{dial} =[0 5], x_{input} = 0, and F_{weight} = [0 300].

STEP 4 – Synthesize Actual Components Using I-Agents:

For each completed alternative configuration, instantiation agents (I-Agents) replace the functional components with actual catalogue components that may be purchased or fabricated. The I-Agents perform the replacement operations through the use of the extracted system equations. I-Agents choose preferred actual components using different preference strategies, as a function of the calculated states from the extracted equations.

Application: Figure 9 shows the instantiation of an example configuration for the weighing machine problem.

STEP 5 – Evaluate, Sort, and Modify Designs:

Instantiated configurations are next evaluated and sorted using a strategy based on Pareto-optimization frontiers. The evaluation and sorting operations lead to the categorization of designs into Pareto optimal, good, and bad populations. These categories are determined based on the input specifications. Poor designs are eliminated from the iterative synthesis process, and Pareto optimal and good designs are carried forward for refinement and the synthesis of further configurations.

The remaining configurations enter the modification stage of the process. Fragmentation agents (F-Agents) are implemented to improve the states of the configurations. These agents selectively remove and add components to configurations based on a number of preference strategies. Components that don't affect the system states (or negatively affect the states) are early candidates for removal. Function sharing is currently not part of this modification approach, but it is viewed as an important research extension.

STEP 6 – Repeat Process to Termination with M-Agents:

Manager agents (M-Agents) monitor the emerging configurations and control the overall process. These agents control other agents and the design configurations through the Pareto optimality strategy. These agents also group component chunks (subsystems of modules) that are successful at meeting the input specification. These chunks are used in constructing new alternatives as the process continues. The process terminates when no improvement is seen along the Pareto frontier of the design space.

Application: Figure 9 shows a Pareto design that emerges from the A-Design process. The design objective values, calculated for the configuration, are shown in the figure, in addition to the component descriptions.

V. DISCUSSION OF THE STATE-OF-THE-ART: Golden Nuggets

Considering the body of function-based synthesis research, the methods, their analysis, and their applications demonstrate compelling results. Concepts may be generated from formal methods, beginning with a functional description or specification. The generated concepts are intriguing. They follow fundamental theories of physics and design. They are also innovative and broad in number and in scope.

On the "flip side," function-based synthesis methods have only considered relatively small-scale applications and only a fraction of the possible set of customer needs. They are currently not used widely in industrial practice; they do not show great coverage of our current function vocabulary or design process, and they are far from the generation of realizable geometries and structural topologies.

Even with these limitations, however, great strides have been taken in the research field. We can learn much from this state-of-the-art. Let's consider a brief summary of the important features of the current research. This summary is based on observations from the research review and technical descriptions, where no particular order or priority is implied.

- Functional -description or functional-specification languages drive function-based synthesis methods. Further research is needed to determine preferred types of representations, especially as more functions of electro-mechanical designs are used in the synthesis process. Hybrid representations, integrating both descriptions (function-basis language) and specifications have yet to be investigated.
- Search techniques include exhaustive, agent-based, set-based, and optimization (or a hybrid).
- Embodiment techniques should be advanced as part of the synthesis process. Few methods consider the generation of detailed geometry and topology. Perhaps a hybrid of shape grammars and function-based synthesis methods will help to achieve this capability.
- The methods studied in this chapter are just beginning to be tested with industrial type problems, such as active dynamic systems (vehicle design), and MEMS components. This component of the research model (Figure 3) must be explored further. Methods must be tested in industry, and, in parallel, descriptive studies in industry must be carried out to enter the research activities through the "Need Based

on Observation and Analysis of Industry" portal. Currently, all of the synthesis methods (in this chapter) began as "New Idea…" work, focusing on basic research issues.

- Agent-based, grammars, and other strategies exist for the synthesis and control activities of the method architecture (Figure 2). Comparisons of these techniques are needed to determine their scope for formal representations.
- Synthesis methods are now exploring the ability for component integration and function sharing as part of the synthesis process. This area of research must continue to aid the efficacy of the methods for industrial practice.
- Synthesis methods are useful even if actual geometry is not generated as an output. Most of the Dynamic Systems methods create abstractions as alternative designs. It is left to the designer to interpret the results and embody the concepts. This process is nonlinear and not unique (Otto and Wood, 2000), but it does greatly assist designers in creating a broad range of creative solutions.
- User interface issues are not treated in any of the work reviewed thus far. As the research transforms or develops from methods to tools, this issue will become paramount. It may even "make or break" a given method's use in practice.
- Algorithmic approaches to design, as reviewed in this chapter, are generally at the concept design level.
- Issues of ergonomics are not treated by the "automated" design methods found in this review. This result is similar to the mechanical design executed at the beginning of the industrial revolution, where machine function was the dominant factor and human-machine-interface functions were not given much if any consideration. A review of the dominant functions, with respect to customer needs, shows that ergonomic-type functions, such as "import/export hand," are the most critical to achieve customer satisfaction (Little and Wood, 1997; Stone and Wood, 1999).

Building on this thought, an important fraction of typical customer needs (or classes of customer needs) are covered by current function-based synthesis methods. This fraction focuses on the physics and functions related to power flow, in addition to limited kinematics. While this fraction is important, many other types of customer needs are apparent in all devices and products (Urban and Hauser, 1993; Otto and

Wood, 1998; McAdams, *et al*., 1999). The synthesis research field will need to grow to subsume these addition types of customer needs and the functions they generate in a design.

- Many technologies will converge to standardized organs (architectures), such as motors, fasteners, etc. Catalogue design methods should be able to exploit this principle of electro-mechanical design in future research.
- Common functional languages (descriptive or specifications) require further research. Through common/formal languages, repositories of functional and structural elements may be created, filtered, and shared between the competing synthesis methods.
- Scale is a critical and unresolved issue of the research field. All of the methods have only considered relatively small-scale design problems. What will happen as the scale increases, e.g., from household consumer products to medium-scale power equipment (lawnmowers, etc.) or large-scale products such as automobiles and aircraft?
- The current function-based synthesis methods focus on component combinations with a serial type of architecture. What about other types of architecture, such as the various modular types, integral, mass customization, tunable, etc. (Otto and Wood, 2000; Stone, *et al*., 2000a,b)? The research in product architecture has advanced significantly over the last decade. Synthesis methods should integrate these research advancements and formalisms.
- Formal engineering design synthesis has been compared to the formalisms and level-of-automation in VLSI design (Calvez, *et al*., 1993; Sander and Jantsch, 1999). In particular, a debate exists regarding the analogous possibilities of compilation in mechanical design with that of VLSI design (Antonsson, 1997; Whitney, 1996). No attempt is made in this chapter to answer, explicitly, the fundamental issues in this debate. It is clear, however, that new and significant data exists from recent synthesis methods (in the mechanical and electro-mechanical domains). These data, while far from complete, should be strongly considered in reviewing the opinions of Antonsson and Whitney.

- Design principles and theories underlie the synthesis and control strategies of the methods reviewed here. Research into principle development (for specific domains) will greatly assist the future development of the synthesis methods. Better formalisms and representations will create better methods.

VI. ANALYSIS FINALE: AN EXAMPLE VISION FOR THE FUTURE

The field of function-based synthesis has advanced significantly over the last decade, based on the contributions of a number of talented researchers around the world. The analysis and discussion in the previous sections provide many points/opinions regarding the future of function-based synthesis. In particular, the most important point concerns the research model (Figure 3). Descriptive industrial studies (to create new research needs) and industrial tests are needed to advance the field to the next stage. In the subsections below, an example approach to this point is described. This approach might be used as an analogy for similar efforts.

Walk Before We Run

In order for function-based synthesis to gain industry acceptance, successes must be demonstrated in a real engineering design setting. One method of achieving this goal is to develop a full-scale working tool that embodies one of the methods reviewed in this chapter. Unfortunately, none of the present set of methods is ready for full embodiment and application in industry. The research results just do not appear to be mature enough to move forward. A more appropriate approach may be to start with a much more focused method and embody that method for use by industry. The benefits of this approach are multifaceted. The first benefit is that valuable descriptive information can be gathered on the effectiveness of the method in an industrial setting thus leading to refinement of the method and establishment of the feedback mechanism discussed in the research model (Figure 3). A second and equally important benefit is the establishment of credibility for this type of design tool with industry. These methods must be shown to be either advantageous over or synergistic with traditional approaches to design; otherwise, they will remain academic exercises without industrial relevance.

An example of a generative method that takes this focused approach is *effort flow analysis* (EFA) presented by Jensen, *et al.* (2000). Effort flow analysis is the embodiment of a method that is based on descriptive analysis of an industrial need. The method evolved out of industrial design problems where the goal was to redesign automotive components to reduce complexity. The EFA method has been applied to the redesign of an auxiliary sun visor, for example, for a major automotive manufacturer with dramatic bottom line results. An overview of the technique is presented in the following section.

Effort Flow Analysis

Effort flow analysis, as presented here, is a redesign method. As a redesign method, EFA starts with an existing product with an established set of customer needs, a known functional model (function structure), and an existing topology for the artifact. The goal of the method is to reduce product complexity thus leading to a reduction in production costs. In this case, product complexity is reduced through the synthesis of function sharing within the system. EFA highlights opportunities for function sharing by developing a diagram depicting the flow of effort as it is transmitted through the components that embody the known functional model. This diagram is called, appropriately, an effort flow diagram and represents the workframe of the method. Effort flow diagrams represent components using nodes (symbolized by circles or squares), and the interfaces between components are drawn as links (directed arrows) connecting the components. The links are the critical element, as it is through the interface where the effort flow takes place. Each of the links is then characterized based on the presence of relative motion (R-Link) at the interface represented by the link. The relative motion in the link has been characterized at three levels:

- Class I; A link used to represent effort flow between components having relatively small ratios of displacement to characteristic dimension. A characteristic dimension, as noted here, is a distinguishing spatial property necessary to achieve the R-Link function between components. A small ratio of displacement to characteristic dimension implies that the resulting strains and associated stresses are small enough to keep the members in the elastic region over multiple actuations.
- Class II; A link used to represent effort flow between components that perform unique or distinct material functions that must be maintained.
- Class III; A link used to represent effort flow between components that have relatively large ratios of displacement to characteristic dimension or that must perform multiple functions.

The most obvious opportunities for component combination are made apparent by looking for links where there is no relative motion, but a more sophisticated approach involves the analysis of the links that do have relative motion. These links present the opportunity for novel designs based on the generation of new hybrid components that satisfy the original functional and geometric requirements of the device. Effort flow diagrams also seek to represent general topological arrangement of the components so there is a geometric component to the diagram as well as a functional component. The existing geometry of the artifact must be considered as function sharing is synthesized. The goal is to generate combined components that interface with the existing geometry, thus reducing the impact on the overall system design. An extension to this method, that will lend itself to computational application, is the automated generation of geometric forms that satisfy the geometric constraints while combining components through a link with relative motion. The method architecture for this approach is shown in Figure 23. Finite element analysis is a likely candidate tool for this application. Examples of generating geometry can be found in the compliant mechanism literature (Nishiwaki,, *et al*., 1998; Frecker, *et al.,* 1997).

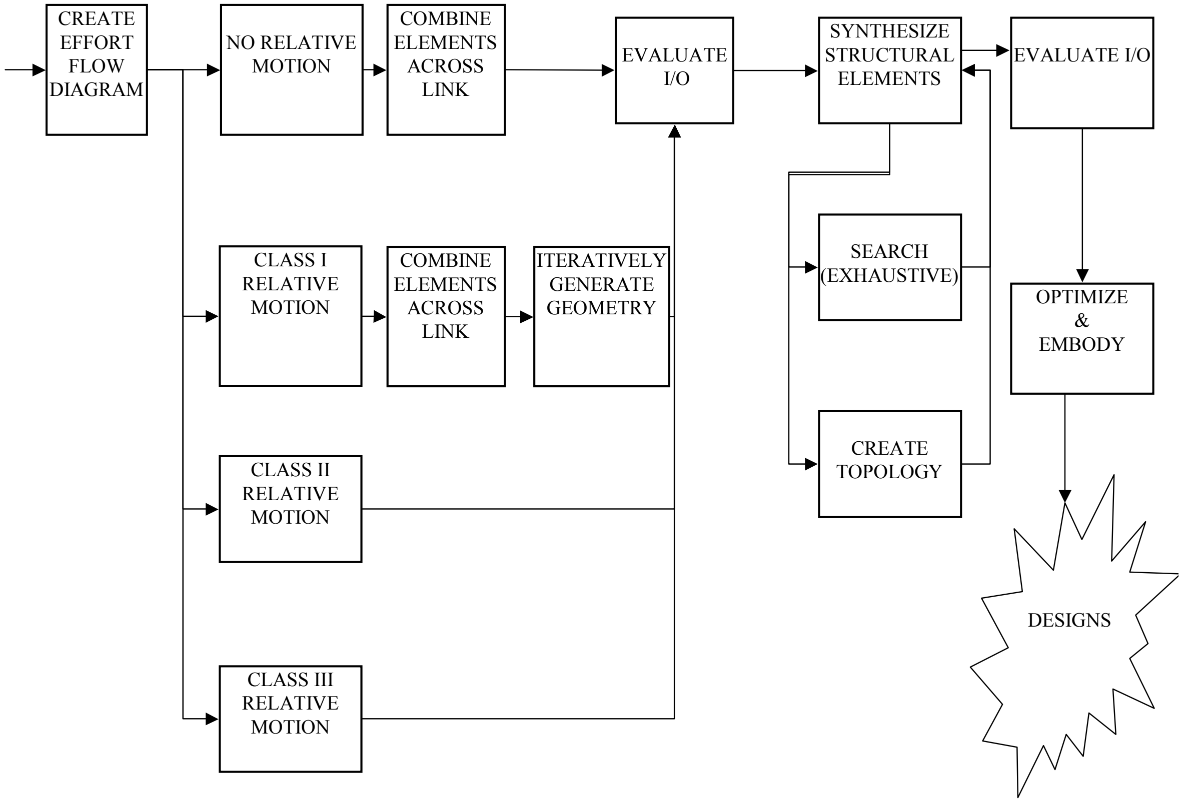


Figure 23. Method Architecture for EFA synthesis.

VII. CLOSURE

Concept generation represents the time when product function and architecture are transformed to actual geometry. This stage in product development is exciting and challenging. It is the time when creativity and design principles are used to create innovative solutions. It is also the time when the first glimpses of a realized product appear from the design teams' inner thoughts and dreams.

Formal function-based synthesis methods are emerging as a potential tool set for concept generation. These methods currently focus on the areas of dynamic systems, agent-based, and catalogue design synthesis (Figure 4). Within these areas, significant advancements have been made, demonstrating basic research results as well as early applications.

Considering the household consumer products in Figure 24, the current methods are not capable of generating the full set of functions or embodiments shown in these products. Skeletal structural elements or abstract functional representations of such products are the state-of-the-art.

This limitation should not deter us, however. For illustration purposes, Figure 25 shows two versions of a concept for a printer product. The left-hand figure is a mechanical breadboard prototype of the product. The right-hand figure shows the production version of the product. Based on the analysis of this chapter, the prototype product, shown in Figure 25, could very likely be generated with the current methods. While many functions and embodiment issues must be solved to arrive at the production version of the product, the abilities to generate a design like the prototype is significant and exciting. What was a mere dream ten years ago is realizable today!!

Figure 24. Consumer product examples (Otto and Wood, 2000).

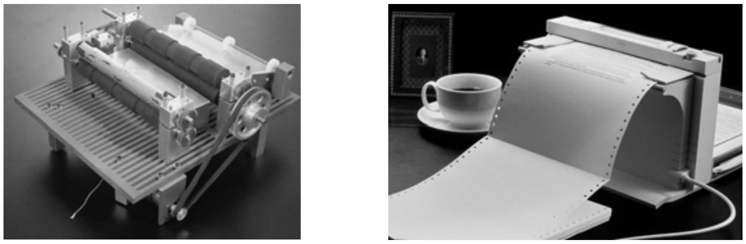


Figure 25. Mechanical breadboard prototype and a production version of a printer product (Otto and Wood, 2000).

ACKNOWLEDGEMENTS

The authors offer a special “thank you” to Jon Cagan, Matthew Campbell, Erik Antonsson, Tom Connolly, and Raul Longoria for their advice, contributions of knowledge, and investment of time to discuss their research. The work reported in this document was made possible, in part, by various grants from the National Science Foundation. The authors also wish to acknowledge the support of the United States Air Force Academy, Fluor-Daniel, Ford Motor Company, Texas

Instruments, Desktop Manufacturing Corporation, and the UT June and Gene Gills Endowed Faculty Fellow. Any opinions, findings, or recommendations are those of authors and do not necessarily reflect the views of the sponsors.

REFERENCES

Antonsson, E. K., 1987, "Development and Testing of Hypotheses in Engineering Design Research," ASME *Journal of Mechanisms, Transmissions, and Automation in Design* (currently the *Journal of Mechanical Design*), Vol. 109, pp. 153-154.

Antonsson, E. K., 1997, "The Potential for Mechanical Design Compilation," *Research in Engineering Design*, Vol. 9, pp. 191-194.

Baher, H., 1984, *Synthesis of Electrical Networks*, John Wiley & Sons, New York.

Bradley, S. R. and Agogino, A. M., 1994, "An Intelligent Real Time Design Methodology for Component Selection: An Approach to Managing Uncertainty," *Journal of Mechanical Design*, Vol. 116, pp. 980-988.

Blessing, L.T.M, Chakrabarti, A., Wallace, K.M., 1998, "An Overview of Descriptive Studies in Relation to a General Design Research Methodology", In *Designers: The Key to Successful Product Development*, Springer-Verlag, Munich, Heidelberg, New York, 1998.

Brune, O., 1931, "Synthesis of a Finite Two-Terminal Network Whose Driving-Point Impedance is a Prescribed Function of Frequency," *MIT Journal*.

Carlson-Skalak, S., White, M. D., and Teng, Y., 1998, "Using an Evolutionary Algorithm for Catalog Design," *Journal of Research in Engineering Design*, Vol. 10, pp. 63-83.

Calvez, J. P., Heller, D., and Bakowski, O., 1993, "Functional-Level Synthesis with VHDL," *IEEE*, Paper No., 0-8186-4350-1/93, pp. 554-559.

Campbell, M. I., Cagan, J., and Kotovsky, K., 1998, "A-Design: Theory and Implementation of an Adaptive, Agent-Based Method of Conceptual Design," In Artificial Intelligence in Design '98, J. Gero and F. Sudweeks, eds., Lisbon, July 20-23, pp. 579-598.

Campbell, M. I., Cagan, J., and Kotovsky, K., 1999, "An Agent-Based Approach to Conceptual Design in a Dynamic Environment," Journal of *Research in Engineering Design*, Vol. 11, pp. 172-192.

Campbell, M. I., Cagan, J., and Kotovsky, K., 2000, "Agent-Based Synthesis of Electromechanical Design Configurations," ASME *Journal of Mechanical Design*, Vol. 122, pp. 1-9.

Cantamessa, M., 2000, "Design Methods and Tools with Respect to Industrial Practice," Presentation at *WDK Design Society Workshop*, Rigi, Switzerland, March 23, 2000. (Contact: Marco Cantamessa, Politecnico di Torino, Italy.)

Cauer, W., 1958: *Synthesis of Linear Communication Networks*, Translated from the German Kanausenberger, G.E., and Warfield, J.N., eds., McGraw Hill, New York, NY.

Chakrabarti, A. and Bligh, T. P., 1994, "An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part I: Introduction and Knowledge Representation," Journal of *Research in Engineering Design*, Vol. 6, pp. 127-141.

Chakrabarti, A. and Bligh, T. P., 1996a, "An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part II: Kind Synthesis," Journal of *Research in Engineering Design*, Vol. 8, pp. 52-62.

Chakrabarti, A. and Bligh, T. P., 1996b, "An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part III: Kind Synthesis," Journal of *Research in Engineering Design*, Vol. 8, pp. 116-124.

Chakrabarti, A. and Bligh, T. P., 1996c, "An Approach to Functional Synthesis of Mechanical Design Concepts: Theory, Applications, and Emerging Research Issues," *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, Vol. 10, pp. 313-331.

Connolly, T. J., and Longoria, R.G., 2000, "A Simulation-Based Synthesis Method for Engineering Systems That Contain Active Elements," *Proceedings of the 2000 ASME International Mechanical Engineering Congress and Expo*: ASME Symposium on Automated Modeling, November 5-10, Orlando, Florida.

Connolly, T. J., 2000, "Synthesis of Multiple-Energy Active Elements for Mechanical Systems," Ph.D. Dissertation, The University of Texas at Austin, May 2000.

Duffy, A.H.B., O'Donnell, F.J., 1998, "A Model of Product Development Performance," In *Designers: The Key to Successful Product Development*, Springer-Verlag, Munich, Heidelberg, New York, 1998.

Finger, S. and Rinderle, J., 1989a, "A Synthesis Strategy for Mechanical Devices," *Research in Engineering Design*, Vol. 1, No. 2, Spring-Verlag, New York.

Finger, S. and Rinderle, J. R., 1989b, “A Transformational Approach for Mechanical Design Using a Bond Graph Grammar,” *Proceedings of the ASME Design Theory and Methodology Conference*, Montreal, Canada, September, pp. 107-116.

Foster, R., 1924, “Theorems on the Driving Point Impedance of Two-Mesh Circuits,” *Bell Systems Technical Journal*, Vol. 2, p 651.

Foster, R., 1924, “A Reactance Theorem,” *Bell System Technical Journal*, Vol. 3, pp. 259-267.

Frecker, M.I., Ananthasuresh, F.K., Nishiwaki, S., Kikuchi, N., Kota, S., 1997, “Topological Synthesis of Compliant Mechanisms Using Multi-Criteria Optimization,” ASME *Journal of Mechanical Design*, Vol. 119 No. 2, pp 238-245.

Hoover, S. P. and Rinderle, J. R., 1989, *A Synthesis Strategy for Mechanical Devices, Research in Engineering Design*, Vol. 1, pp. 87-103.

Hubka, V., 1982, *Principles of Engineering Design*, Butterworths, London.

Hubka, V. and Eder, W. E., 1988, *Theory of Technical Systems*, Springer-Verlag, Berlin.

Hubka, V., Andreasen, M.M., and Eder, W.E., 1988, *Practical Studies in Systematic Design*, Butterworths, London.

Hubka, V., Eder, W.E., 1998, “Theoretical Approach in Design Methodology,” In *Designers: The Key to Successful Product Development*, Springer-Verlag, Munich, Heidelberg, New York, 1998.

Iyengar, G., Lee, C., Kota, S., 1994, “Towards an Objective Evaluation of Alternate Designs,” ASME *Journal of Mechanical Design*, Vol. 116, No 2, June, pp. 487-492.

Jensen, D.L., Greer, J.L., Wood, K.L., Nowack, M.L., 2000, “Force Flow Analysis: Opportunities for Creative Component Combination,” *Proceedings of the 2000 ASME International Mechanical Engineering Congress and Expo*., November 5-10, Orlando, Florida.

Kannapan, S. and Marshek, K. M., 1987, “Design Synthetic Reasoning: A Program for Research,” Dept. of Mechanical Engineering, The University of Texas at Austin, Technical Report 202.

Kannapan, S. M. and Marshek, K. M., 1991, “Design Synthetic Reasoning: A Methodology for Mechanical Design,” Research in Engineering Design, Vol. 2, pp. 221-238.

Karnopp, D.C., Margolis, D.L., & Rosneberg, R.C., 1990, *Systems Dynamics: A Unified Approach*, Second Ed., Wiley Interscience, New York.

Kota, S. and Chiou, S. J., 1992, "Conceptual Design of Mechanisms Based on Computational Synthesis and Simulation, *Research in Engineering Design*, Vol. 4, pp. 75-87.

Krumhauer, P., 1974, "Rechnerunterstützung für die Konzeptphase der Konstruktion," PhD Thesis, TU Berlin, Berlin, Germany.

Little, A.D. and Wood, K.L., 1997, "Functional Analysis: A Fundamental Empirical Study for Reverse Engineering, Benchmarking, and Redesign," *ASME Design Theory and Methodology Conference*, Sacramento, CA, CD-ROM, DETC97/DTM-3879.

Malmqvist, J., 1994: "Computational Synthesis and Simulation of Dynamic Systems," ASME Proceedings of the *Design Theory and Methodology Conference*, DTM '94, Minneapolis, MN, Sept. 11-14, pp. 221-230.

McAdams, D. A., Stone, R. B., and Wood, K. L., 1999, "Functional Interdependence and Product Similarity Based on Customer Needs," *Journal of Research in Engineering Design*, Vol. 11, No. 1, pp. 1-19.

Navinchandra, D., Sycara, K., Narasimhan, S., 1991, "A Transformational Approach to Case-Based Synthesis," *AI-EDAM*, Vol. 5, pp. 31-45.

Nishiwaki, S., Frecker, M.I., Min, S., Kikuchi, N., 1998: "Topology Optimization of Compliant Mechanisms Using The Homogenization Method," *Int. J. Numer. Meth. Eng.*, Vol. 42, pp 535-559.

Otto, K.N. and Wood, K.L., 1998, "Product Evolution: A Reverse Engineering and Redesign Methodology," Journal of *Research in Engineering Design,* Springer-Verlag, London Limited, pp 226-243.

Otto, K.N., Wood, K.L., 2000, *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall, NY.

Pahl, G. and Beitz, W., 1996, *Engineering Design: A Systematic Approach*, 2nd Edition, Springer-Verlag, London Limited

Pahl, G, 1998, Keynote, In *Designers: The Key to Successful Product Development*, Springer-Verlag, Munich, Heidelberg, New York, 1998.

Paynter, H. M., 1961, *Analysis and Design of Engineering Systems,* The MIT Press, Cambridge, MA.

Prabhu, D.R., and Taylor, D.L., 1988, "Some Issues in the Generation of the Topology of Systems with Constant Power-Flow Input-Output Requirements," Advances in Design

Automation – 1988, Proceedings of the 1988 ASME Design Automation Conference, DE Vol 14, Kissimmee, FL, September 25-28, pp. 41-48.

Prabhu, D.R., and Taylor, D.L., 1989, “Synthesis of Systems From Specifications Containing Orientations and Positions Associated With Flow Variables,” Advances in Design Automation – 1989, Proceedings of the 1989 ASME Design Automation Conference, DE-Vol. 19-1, Montreal, Quebec, September 17-21, pp. 273-280.

Redfield, R.C., and Krishnan, S., 1991, ‘Automated Conceptual Design of Physical Dynamic Systems Using Bond Graphs,” Proceedings of 1991 *International Conference on Engineering Design* (ICED '91), Zurich, Switzerland.

Redfield, R.C., & Krishnan, S., 1993, “Dynamic System Synthesis With a Bond Graph Approach: Part I - Synthesis of One-Port Impedances,” ASME *J. of Dynamic Systems, Measurement, and Control*, Vol. 115, No. 3, pp. 357-363.

Redfield, R.C., 1993, “Dynamic System Synthesis With a Bond Graph Approach: Part II - Conceptual Design of an Inertial Velocity Indicator,” ASME *J. of Dynamic Systems, Measurement, and Control*, Vol. 115, No. 3, pp. 364-369.

Redfield, R.C., 1999, “Bond Graphs in Dynamic Systems Design: Concepts for a Continuously Variable Transmission,” 1999 *International Conference on Bond Graph Modeling and Simulation* (ICBGM '99), The Society for Computer Simulation, Vol. 31, No. 1, pp. 225-230.

Sander, I. and Jantsch, A., 1999, “Formal system Design Based on Synchrony Hypothesis, Functional Models, and Skeletons,” Proceedings of the 12th *International Conference on VLSI Design*, IEEE Paper No. 0-7695-0013-7/99, January, pp. 318-323.

Schmidt, L. and Cagan, J., 1995, “Recursive Annealing: A Computational Model for Machine Design”, *Research in Engineering Design*, Vol. 7, No. 1, Spring-Verlag, New York, pp. 102-125.

Schmidt, L. and Cagan, J., 1996, “Grammars for Machine Design,” In Artificial Intelligence in Design, J. Gero and F. Sudweeks, eds., Kluwer Academic, Netherlands, pp. 325-344.

Schmidt, L. and Cagan, J., 1997, “GGREADA: A Graph Grammar-Based Machine Design Algorithm,” Research in Engineering Design, Vol. 9, No 2, Spring-Verlag, New York, pp. 195-213.

Schmidt, L. and Cagan, J., 1998, “Optimal Configuration Design: A Integrated Approach Using Grammars,” ASME *Journal of Mechanical Design*, Vol. 120, No 1, pp. 2-9.

Schmidt, L., Shetty, H. and Chase, S., 1998, "A Graph Grammar Approach for Structure Synthesis of Mechanisms," Proceedings of the 1998 ASME Design Engineering Technical Conferences, DETC98/DTM-5668, Atlanta, GA, Sept. 13-16.

Schregenberger, 1998: "The Further Development of Design Methodologies," In *Designers: The Key to Successful Product Development*, Springer-Verlag, Munich, Heidelberg, New York.

Stone, R.B. and Wood, K.L., 1999, "Development of a Functional Basis for Design," *Proceedings of DETC99*, 1999 ASME Design Engineering Technical Conferences, DETC99/DTM-8765, Las Vegas, NV. (Also, to appear in the ASME *Journal of Mechanical Design*.)

Stone, R. B., Wood, K. L., and Crawford, R. H., 2000a, "A Heuristic Method to Identify Modules from a Functional Description of a Product," *Design Studies*, Vol. 21, No. 1, pp. 5-31, January.

Stone, R. B., Wood, K. L., and Crawford, R. H., 2000b, "Using Quantitative Functional Models to Develop Product Architectures," *Design Studies*.

Tay, H.E., Flowers, W., Barrus, J., 1998, "Automated Generation and Analysis of Dynamic System Designs," *Research in Engineering Design*, Vol. 10, No. 1, Springer-Verlag, New York, NY.

Ullman, D.G., 1993, "A New View of Function Modeling," *International Conference on Engineering Design*, ICED '93, The Hague, August 17-19.

Ullman, D. G., 1997, *The Mechanical Design Process*, McGraw-Hill, NY.

Ulrich K.T., 1988, Computation and Pre-Parametric Design, MIT Artificial Intelligence Lab Technical Report 1043.

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

Urban, G. and Hauser, J., 1993, *Design and Marketing of New Products*, Prentice Hall, New York, 2nd edition.

Ward, A. C., 1988, "A Theory of Quantitative Inference Applied to a Mechanical Design Compiler," Ph.D. Thesis, MIT.

Ward, A. C. and Seering, W., 1989a, "The Performance of a Mechanical Design Compiler," MIT AI Memo No. 1084.

Ward, A. C. and Seering, W., 1989b, "Quantitative Inference in a Mechanical Design Compiler," MIT AI Memo No. 1062.

Welch, R. V. and Dixon, J. R., 1992, "Representing Behavior and Structure During Conceptual Design," *Proceedings of the ASME Design Theory and Methodology Conference*, pp. 11-18.

Welch, R. V. and Dixon, J. R., 1994, "Guiding Conceptual Design Through Behavioral Reasoning," *Research in Engineering Design*, Vol. 6, pp. 169-188.

Whitney, D., 1996, "Why Mechanical Design Cannot Be Like VLSI Design," *Research in Engineering Design*, Vol. 8, pp. 125-138.

BIOGRAPHIES

KRISTIN L. WOOD

Dr. Kristin L. Wood is currently a Professor of Mechanical Engineering at The University of Texas at Austin and the June and Gene Gillis endowed faculty fellow in manufacturing. Dr. Wood completed his M.S. and Ph.D. degrees in Mechanical Engineering at the California Institute of Technology, where he was an AT&T Bell Laboratories Ph.D. Scholar. He received his Bachelor of Science in Engineering Science from Colorado State University, May 1985. The current and near-future objective of Dr. Wood's work is to develop design strategies, representations, and languages that will result in more comprehensive design methods and design teaching aids at both the college and pre-college levels.

James L. Greer

James L. Greer is currently a Ph.D. student in the department of mechanical engineering at The University of Texas at Austin. James completed a M.S. at Purdue University, and a B.S. at The University of California at Davis, both in mechanical engineering. James is a former faculty member of the Department of Engineering Mechanics at the U S Air Force Academy, and will return there upon completion of the requirements for a Ph.D.