A CASE-BASED REASONING APPROACH FOR SYNTHESIS OF ELECTRO-MECHANICAL DEVICES USING BOND GRAPHS

T.N. Madhusudan[1] **Katia Sycara** **D. Navin-Chandra**
Robotics Institute,
Carnegie-Mellon University
Pittsburgh, Pennsylvania, 15213,
USA.

ABSTRACT

Developing a computational representation for electro-mechanical devices that supports *conceptual synthesis* of complex devices from simpler subsystems both in terms of *device behavior* and *form* has been of serious interest. Proposed approaches in literature consider either the synthesis of device behavior or only of shape depending on the limitations of the device model and computational approaches used such as rule-based systems, constraint satisfaction, qualitative reasoning and graph grammars. This paper suggests an approach for electro-mechanical device synthesis based on *Bond graph* models of devices whereby problems of representational ambiguity and incompleteness are overcome. We describe the utility of bond graphs as a comprehensive device model that enables synthesis of physically feasible devices, integrates parametric information regarding form and materials and provides a convenient link to device validation using simulation. A *Case-based design* scheme for behavioral synthesis of electro-mechanical systems is described. We illustrate the process of *elaboration* whereby design cases are retrieved and composed into composite systems with an example. We also suggest how this approach may provide a convenient framework to integrate conceptual, parametric and configuration design in a consistent manner.

Keywords: Bond graphs, Conceptual design, Case-based reasoning

1 INTRODUCTION

Research in design theory and methodology has focused on developing a computational representation of devices that may support algorithmic procedures for the synthesis of complex electro-mechanical systems from simpler components. Thus far we have recognized the role of shape, behavior and material characteristics in determining the design of a product but do not have a comprehensive framework wherein these device attributes can be reasoned about in a *computationally consistent* manner. Lack of models that capture the interaction of design choices regarding form and material attributes with the final required functionality and behavior of a device has been a primary bottleneck. Design tools that deal with specific design issues such as material choices, structural analysis, lumped and distributed parameter analysis exist as islands of design activity without an overall computational model both of the product (integrating all its attributes) and a process of design. Without these product and design process models, it has been extremely difficult to understand and develop tools that systematize and integrate conceptual, parametric and configuration design.

Synthesis of products wherein components are first chosen based on their behavioral characteristics, followed by choices of parametric attributes and then choices of shape and configuration without violating the previous design decisions is an approach with veritable success in the field of VLSI-CAD. Attempts to adapt such an approach to electro-mechanical design have been limited primarily for two reasons, namely, (1) The nominal behavior of mechanical devices is continuous, deals primarily with the transfer of power and only analog models are applicable and (2) The nominal behavior is largely affected by the shape of the com-

[1] This author receives the correspondence concerning this paper, unless otherwise specified.

ponents, their material properties and the physical form of the components. Device models proposed in the DTM literature have attempted to model electro-mechanical devices as devices with *state* in the sense of digital logic using logic-theoretic representations such as graph grammars, influence diagrams, predicate logic etc. It seems that these device representations are more useful for modeling mechanical devices that do show a state-like behavior such as an indexing mechanisms or a mechanical switch but not for modeling a four-bar mechanism or a gear-transmission. Further, the aim at large from a mechanical design viewpoint has been to synthesize shape and form first and then verify if the device provides the requisite behavior. The reason for such an approach is rather obvious because *components* in the mechanical engineering world are primarily identified based on geometry and really not on behavior. This is in contrast to electronic devices since all devices are distinguished primarily on their behavior (logic) and all seemingly have the same form and also devices share a common process of manufacture. It is relevant to note that a bottleneck similar to electro-mechanical design exists in the development of synthesis tools for *analog* electrical circuit design. This is because physical reality is ultimately continuous and one needs to model this continuum to predict real-world device behavior.

This paper outlines a comprehensive representation of electro-mechanical devices using bond graphs. *Bond graphs* [1, 2] view electro-mechanical device behavior from the standpoint of *energy continuity and power balance.* Systems and subsystems are classified both with respect to the number of energy ports through which energy is exchanged and also in terms of the particular internal power transformations involved. The representation recognizes the distinct role of power and signal devices in electro-mechanical devices. Signal flow is considered as a form of low-power-level directional interaction. The bond graph based model provides behavioral building blocks at the analog level and also provides a consistent link to represent form and material information. Though bond graphs are well-known to the controls and analysis community, the utility of bond graphs as a good device model and representation for design has largely been unexplored except for [3, 4]. It is interesting to note that the use of a bond graph based scheme to develop a unified engineering database and its ability to support development of design tools is suggested in [5].

Further, we believe that a synthesis process that proceeds from behavioral synthesis to shape and form may provide a more fruitful approach in the effort to formalize the design process and building tools to support the same. The rationale behind such a viewpoint is that constraints on behavior can be well-defined and formalized and the search of designs guided better in comparison to constraints on shape and geometry. We use the Case-based design[6, 7, 8] computational framework. Case-based design is a computational paradigm that aims at reusing previous designs archived in a repository with well-defined design principles to generate new designs given specifications. Implementation of a case-based design system requires a case representation of electro-mechanical devices, indices for retrieving these devices from the casebase and algorithms whereby these cases can be composed or adapted to generate new solutions.

Our case-representation of devices is based on the bond graph language. Modeling devices in the bond graph language enables one to formulate the synthesis and design of electro-mechanical systems as the *selection and interconnection of a set of standard multiport devices* to meet given specifications. In our case-based representation and synthesis procedures, we use the bond graph based vocabulary to develop indices for design cases based on energy domains of ports, device relationships between ports and interconnectivity topology of subsystems. The bond graph language provides the ability to *snippet* subsystems from complex systems and reuse these subsystems in other devices in a principled manner. We do not address the synthesis of each singleton component but rather the issue of useful assemblies of these components. Specifications for case-based synthesis are provided in terms of the time histories of input and output variables for the requisite physical system and their respective energy domains. The synthesis process generates a variety of interconnective topologies of cases that can transform the given inputs into requisite outputs. Each combination of cases is consequently verified based on the relationships between ports. In this way, viable systems are selected.

Our proposed approach to using bond graphs is in contrast to [3, 4] mentioned earlier. In [3], the representation uses the symbolic device definitions of bond graphs such as resistors, gyrators etc. as vocabulary elements of a behavioral grammar and transformation rules that combine these elements as a grammar sequence are defined. However, the representation does not distinguish between devices that may be represented by a single grammatic term such as all devices that function as transformers but with different input-output power transforming relations. In [4], bond graph elements are viewed as analogous to primitives that provide computational functions such as integration, differentiation, summation etc.; given the differential equation of the device to be synthesized as a specification, an analog computer equivalent is generated. Both approaches have limited themselves to using bond graph elements as abstract computational primitives. Instead, we have recognized that the bond graph formalism is a convenient bridge between qualitative and quantitative representation of physical devices. In contrast to the above-mentioned approaches our use of bond graphs as a language explicitly models the time variation of inputs and outputs of an electro-mechanical system. This is necessary since behavioral design specifications of an electro-mechanical system are provided in terms of nominal time histories of input and output variables. Further, we consider and explicitly represent the detailed relationships that model power and signal transformations between ports of a device and use these relations as part of the case indexing scheme. Combined with the explicit representation of time, the representation of these detailed device relations allows the composition of device subsystems taking into account *time varying* device behavior. This enables composition of devices from subsystems and their consequent verification.

As has been observed in the above-mentioned literature, an obvious limitation of bond graphs is their lumped parameter nature to capture form and material effects and thus the inability to synthesize geometry. In our approach, we do not attempt to generate shape but each case in the casebase has information regarding the geometry of the different components that have been combined behaviorally. Thus when design alternatives are generated, our approach generates physically valid alternatives and also provides information on the various geometries of components that are involved. In our scheme, we restrict our attention to electro-mechanical devices that provide their overall functionality due to their *dynamic* (time-varying) behavior. Our approach does *not* address the issue of synthesizing static structures such as housings, support frames etc. Further, the devices are modelled for their ideal, lossless behaviors and multiple energy domain interactions (such as thermo-fluids) in a device are not considered.

In the following section, we provide a systemic framework to define the problem of synthesis of electro-mechanical devices. In Section 3, we provide a brief review of bond graphs and describe its uses from the viewpoint of developing computational models of synthesis. We describe the process of case-based design with an example and illustrate a taxonomy of electro-mechanical devices that provides a convenient indexing scheme to retrieve electro-mechanical devices in Section 4 and in Section 5 we provide some ideas to extend this model further and concluding remarks.

2 A SYSTEMIC FRAMEWORK FOR BEHAVIORAL SYNTHESIS OF ELECTROMECHANICAL DEVICES

Electro-mechanical devices can be envisioned as a coordinated collection of interacting physical systems [9, 10, 11]. A *system* is assumed to be an entity separable from the environment of the system by means of a physical or conceptual boundary. Further, a system may be composed of interacting parts. We model a device as a dynamic system $\mathcal{S}$ as shown in Figure 1.a.

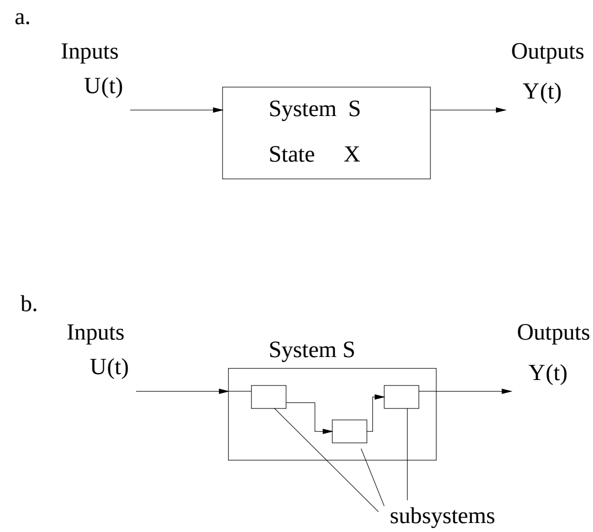

Figure 1. Schematic of a dynamic system S

A system $\mathcal{S}$ is characterized by a set of state variables $\mathcal{X}$ that are influenced by a set of input variables $\mathcal{U}(t)$, that represent the action of the systems environment on $\mathcal{S}$, where t denotes time. The set of output variables $\mathcal{Y}(t)$, are the observable aspects of the system's response or back-effects from the system onto the environment. Further there is a set of intrinsic system parameters $\mathcal{M}$ contained in $\mathcal{S}$. $\mathcal{M}$ depends on the geometry, form and material characteristics of the physical realization of the system $\mathcal{S}$, and in the most general case varies with time. This abstract device model may be used in the following ways:

- *Behavioral analysis.* Given future values of $\mathcal{U}$, present values of $\mathcal{X}$ and the model $\mathcal{S}$, predict the behavior of $\mathcal{Y}$. Assuming that the model of the system is an accurate enough representation of the real system, analysis allows the prediction of the system behavior.
- *Identification.* Given time histories of $\mathcal{U}$ and $\mathcal{Y}$, usually obtained by experiment on real systems, find a model $\mathcal{S}$ and state variables $\mathcal{X}$ which are consistent with $\mathcal{U}$ and $\mathcal{Y}$. Clearly a good model is one that is consistent with a great variety of sets $\mathcal{U}$ and $\mathcal{Y}$.
- *Behavioral synthesis.* Given $\mathcal{U}$ and a desired $\mathcal{Y}$, find S such that $\mathcal{U}$ acting on $\mathcal{S}$ will produce $\mathcal{Y}$. This can be taken as a definition of the behavioral design task. Only in limited contexts are there direct synthesis methods. Often, synthesis is performed via a trial-and-error process of repetitive analysis and identification of a series of candidate systems.
- *Parametric synthesis.* Once a feasible system S is gen-

erated in behavioral synthesis, it is necessary to find values for $\mathcal{M}$, the intrinsic parameters of $\mathcal{S}$ that can accommodate the requisite ranges of values for $\mathcal{U}$ and $\mathcal{Y}$.

- *Configuration synthesis.* If $\mathcal{U}$ and $\mathcal{Y}$ are orientation and position dependent variables (*i.e* vectorial in nature), then ensure that system $\mathcal{S}$ generated in behavioral synthesis satisfies all geometric and topological requirements. This occurs primarily in rotational and translational mechanical systems and it is imperative to maintain orientation and geometric compatibility between the various subsystems that constitute $\mathcal{S}$.

Representing devices involves modeling $\mathcal{S}$ for various types of devices. Results obtained through identification and analysis provide knowledge about a variety of systems $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ and so on. Under this framework, the representation of design specifications addresses the issue of modeling the various types of inputs and outputs of the system. The task of synthesis involves combining a variety of systems $\mathcal{S}_i$ into a complex system $\mathcal{S}$ to produce a desired $\mathcal{Y}(t)$ for an input $\mathcal{U}(t)$. Shown in Figure 1.b is a system $\mathcal{S}$ generated by combining subsystems $\mathcal{S}_i$. The *structure* of a system $\mathcal{S}$ refers to both the subsystems $\mathcal{S}_i$ and their interconnectivity. There is no reference to the real material embodiment i.e shape, size and location of a system when we refer to the structure of a system. We call the material device a *physical realization* of a system $\mathcal{S}$. Each of the systems $\mathcal{S}_i$ may be used in isolation or combinations thereof in the synthesis task to design new complex systems.

The synthesis task is rather complex since *all* physically possible systems have neither been identified nor analyzed. As more complex systems are designed, it is rather imperative that designers be provided with information regarding subsystem behavior and structure of a variety of physical systems identified. Further, tools that provide information on previous designs where similar components and systems have been used can guide the design process. In case-based design, indices for retrieval of systems, subsystems and components from the repository of previous designs are generated based on the design specifications of device input and output (i.e $\mathcal{U}(t)$ and $\mathcal{Y}(t)$); the retrieved systems are combined using domain specific laws and physical principles and validated[12]. Design alternatives thus generated are presented to the user for perusal. Detailed discussion of parametric and configurational synthesis procedures is beyond the scope of this paper.

A number of requirements need to be satisfied by the various elements of the systemic framework described above so that behavioral synthesis can be performed. The essential modeling requirements are:

- *Isomorphism*: Vocabulary used in the representation must be isomorphic to conceptual and physical entities in the real world. This enables the designer to identify, observe and, if possible, measure these entities in a physically realized device. Thus $\mathcal{Y}, \mathcal{U}, \mathcal{M}, \mathcal{X}$ and $\mathcal{S}$ must be well-defined and map onto elements in the physical device.
- *Physical validity*: The representation must adhere to all the laws of physics. Physically impossible devices must not be generated by the synthesis process. The relations between $\mathcal{Y}, \mathcal{U}, \mathcal{M}, \mathcal{X}$ and $\mathcal{S}$ must satisfy the various conservation principles and the valid physical effects that they model.
- *Composability*: For purposes of synthesis the representation must provide the ability to compose primitive systems (components) into complex systems (assemblies) in terms of device behavior. This ensures that different subsystems $\mathcal{S}_i$ can be combined into a complex system $\mathcal{S}$. *The model of a subsystem must characterize the behavior of that subsystem as an entity and independently of how the subsystem is interconnected with other subsystems to form a complex system.* The interaction between subsystems must be well-defined so as to enable composition. From a case-based synthesis perspective, well-defined rules of composition provide the ability to *snippet* physical subsystems from complex systems and reuse them in other devices.
- *Verifiability*: A representation must provide a framework to verify synthesized designs either through domain-dependent rules or simulation. If well-defined analytical models are available, simulation is possible using the system model $\mathcal{S}$ and inputs $\mathcal{U}$ to determine $\mathcal{Y}$. In general, such models are unavailable for a large number of systems, and one has to use the history of success and failure of similar previous designs as guidelines to validate designs.
- *Consistent reasoning across abstraction levels*: To model devices in a comprehensive manner, representations at different levels of abstraction (from behavior to structure) must be used. Design is a task that cuts across all these levels of abstraction. Currently, design representations operate at a single abstraction level. This presents problems of interpretation and translation from one level to the next, which makes the design process more laborious, and possibly introducing inconsistencies. Therefore, a very desirable characteristic of a representational scheme is to allow consistent reasoning across different levels of device modeling.
- *Task independence*: The vocabulary of the representation must be task independent. The same representation may be used for different computational tasks but the semantics of the primitives must not be different. For example a vocabulary that supports simulation

must not be different from a vocabulary that supports synthesis since this prevents the transfer and sharing of knowledge that is generated through computation.

Further, from a computational perspective, the representation must be *tractable.* In our framework, computational tractability means that it must be possible to synthesize a complex $\mathcal{S}$ system from subsystems $\mathcal{S}_i$ in a finite amount of time for a given design specification. If it is not possible to generate a solution, the system must be able to halt the search for a solution. Also, the validation procedure for each design must be well-defined and not computationally intensive. The representation must be *scalable* both in terms of the complexity of the device models $\mathcal{S}$, the complexity of the specifications and the complexity of new designs synthesized. The above-mentioned characteristics are qualitative in nature and difficult to quantify. In order to facilitate case-based design, device models must also provide a rich vocabulary to represent and index devices. The case representation and organization determine the efficiency of retrieval schemes and case synthesis. In the following section, we describe our bond graph based representation for devices.

3 BOND GRAPHS AND DEVICE SYNTHESIS

A device modeling scheme that is physically valid and computationally formalizable is provided by bond graphs. The bond graph formalism provides the necessary conceptual structure to support the systemic framework for synthesis described in the foregoing section. The bond graph model provides a convenient scheme for capturing behavior for individual components and also the behavior of an overall system that is assembled from these components. First we describe the basic characteristics of bond graphs and then consider its implications for synthesis.

The bond graph formalism [2, 13, 14] models devices in terms of the energy flow in the device. The formalism identifies three types of energy interactions among *all* physical devices. The energy behaviors of devices are *energy storage, energy dissipation and energy transmission.* Complex device behavior arises when components with storage, dissipation and transmission behavior are assembled together. The dynamic behavior of physical devices is derived by the application of *instant-by-instant energy conservation.* In the bond-graph formalism, devices are modelled by components connected at places where power can flow between the components. Such places are called *ports* and devices with one or more ports are called *multiports.*

Energy storage and dissipation behavior is exhibited by devices with *one* power port. Devices such as springs, resistors, masses etc. are modelled as one-port devices. *Energy transmission* behavior is exhibited by devices with multiple input and output ports. A T-pipe is modelled as a three-port device. Devices such as motors, slider-crank mechanisms and cams are two-port devices. Power ports allow for power flow between subsystems that are interconnected. In the port-model formalism of bond graphs, each power port of a device has four generalized variables, namely, *effort* $(e(t))$, *flow* $(f(t))$, *effort integral* $(\int e(t)dt)$ denoted as $\mathcal{E}(t)$ and *flow integral* $(\int f(t)dt)$ denoted as $\mathcal{F}(t)$. The *power* $(P(t))$ is equal to $e(t) * f(t)$. The energy flowing through a port over a period of time $E(t)$ is given by $\int e(t).f(t)dt$ or $\int f(t).d\mathcal{E}(t)$ or $\int e(t).d\mathcal{F}(t)$. A power port in a device belongs to an energy domain. Table 1 shows all the possible energy domains and corresponding effort and flow variables of physical systems. The first column lists the energy domain and the ensuing columns the effort, flow, power, effort integral, flow integral and energy variables of that energy domain. Energy domains that involve radiative transfer of energy (solar, light, and radiated heat energy) are not modelled though successful attempts have been made to extend the bond graph methodology to radiative phenomena.

Devices that transform power are modeled in terms of relationships between the effort and flow variables at the power ports. Table 2 shows examples of two-port power devices and the relationships between their ports. The second column lists the relationships in terms of effort and flow variables and the third column gives the example of a device that has the relationship between its input and output ports. For a two-port device, at every instant of time $e_1(t).f_1(t) = e_2(t).f_2(t)$, where the subscript 1 denotes input port and subscript 2 denotes output port. The above equation implies that in a two-port system whatever power is flowing into one side of the 2-port is simultaneously flowing out of the other side. There are numerous other types of device relations possible as long as they satisfy the law of conservation of energy such as relations 5 and 6 in Table 2, where $\mathcal{G}$ and $\mathcal{H}$ denote other functions such as exponential, log etc. Similarly, relations between effort and flow variables can be defined for other multiport systems. The relations between the port variables are obtained via analysis applying the laws of mechanics and electromagnetism, or by experimental testing.

The relationships listed in the Table 2 have algebraic coefficients such as k and g which depend on the geometry and material properties of the device. These are the lumped parameters of the system. For each component device, these lumped parameters need to be identified apriori depending on the physical realization of the device.

The bond graph formalism satisfies all the conservation and thermodynamic laws. Physical laws such as Faraday's law etc. are captured in terms of the relationships between the port variables and thus a convenient abstraction of phys-

Domain	Effort	Flow	Power	Effort Integral	Flow Integral	Energy
Mech. trans.	Force (F)	Velocity (v)	F.v	Momentum	Displacement (x)	Mech. Work (W)
Mech. Rotation	Torque (T)	Angular Velocity (ω)	T.ω	Angular Momentum	Angle (Θ)	Mech. Work
Hydraulic	Pressure (P)	Flowrate ($\dot{Q}$)	$P.\dot{Q}$	Pressure Momentum	Volume (V)	Hydraulic Energy
Electro-Magnetic	Voltage (e)	Current (i)	e.i	Flux (λ)	Charge (q)	Electrical energy
Thermal	Temperature	Entropy-flow	*	*	*	*

Table 1. Energy and Power parameters

ical effects is provided. For example, Faraday's law in a DC motor relates the input voltage (effort) to the output velocity (flow) through the relation $e = Blr\omega$, where e is the voltage, ω is the angular velocity of the output shaft, B is the magnetic field intensity and l and r are geometric parameters.

At this juncture, we contrast analog electrical and mechanical devices in terms of the bond graph language. It is of interest that analog electrical circuits are primarily built from only *passive* one-port devices such as resistors, capacitors and inductors. Passive devices are those that do not contain a power source. *Active* devices such as controlled voltage and current sources and transistors may be multiported but are treated as signal devices. There are many more physically different types of power multiport devices in the mechanical domain than in the electrical domain. Further, mechanical devices involving mechanisms are *spatially* dependent unlike electrical circuits. An electrical circuit may be laid out in any way in space and still function satisfactorily while a mechanism is critically dependent on the geometry and orientation of the linkages. However, a network of hydraulic pipes may function satisfactorily independent of the spatial routing of the pipes. Thus though the bond graph is generalizes across multiple energy domains, device modeling in each energy domain does have its own constraints.

The bond graph formalism provides a comprehensive framework to compose, decompose and reuse power and signal systems and subsystems. The port variables i.e. efforts and flows can be identified and measured in devices satisfying the criteria of isomorphism to reality. Relationships can be identified between port variables in power and signal devices. Further, the device relationships provide models to verify input-output behavior. Thus the *same device representation is used for both synthesis and verification of behavior.* When storage elements and junction devices are included, the bond graph formalism allows generation of systems of differential equations that allows for simulation.

A problem that has been recognized in the literature, but not resolved in a computationally satisfactory manner, is that there is a many-to-many relationship between device input-output behavior and system structure. A given $\mathcal{U}(t) \rightarrow \mathcal{Y}(t)$ transformation can be provided by a variety of systems $\mathcal{S}$, each made up of different combinations of subsystems. Further, a given system $\mathcal{S}$ can transform an infinite number of inputs $\mathcal{U}(t)$ into the corresponding $\mathcal{Y}(t)$. Bond graphs provide a set of behavioral primitives whereby the many-to-many relationship can be represented satisfactorily by representing all *similar* systems in terms of the primitives. Differences between systems can be identified to resolve ambiguity. The many-to-many relationship between input-output behavior and system structure is also compounded by the concern to consider geometry and shape and the physical realization, $\mathcal{P}$, of a system $\mathcal{S}$. A single system $\mathcal{S}$ has numerous physical realizations $\mathcal{P}_i$. For example, a spring-like behavior with specific compliance can be obtained by geometries such as a helical spring, cantilevered beam, a pneumatic cylinder etc. Synthesizing such physical structures computationally may difficult but since they are used from a behavioral view point, the bond graph language provides a consistent indexing scheme to identify and retrieve a variety of physical devices sharing similar functional and behavioral characteristics. Our representation recognizes these different abstractions. A case provides an integrating framework for these different but necessary device abstractions. In addition, the explicit representation of relations includes coefficients, such as k and g which depend on the geometry and material properties of the device, thus providing a consistent indexing mechanism that differentiates among the different abstractions that comprise a comprehensive device model, and at the same time provide a bridging scheme for reasoning between conceptual and parametric design.

Computationally, the bond graph based model provides a well-defined set of behavioral primitives. The synthesis task can be formulated as a combinatorial search problem since subsystems and primitives can be composed in numerous ways to meet a behavioral specification. The graph-based systemic model provides a convenient computational framework where systems can be composed *algorithmically*[15]. The creative aspect in the behavioral synthesis task is generating only feasible candidates from the numerous that are possible. This selective generation process is largely guided by expertise gained in building previous designs and systems and has motivated our efforts to build a case-based design system. In the following section, we describe the representation of electro-mechanical devices as cases and illustrate case-based synthesis.

4 CASE-BASED DESIGN

The general device model described in the previous section provides indices for design cases in terms of *behavior, energy domain of the ports, system structure and the various relations and there combinations thereof.* The power and signal devices can also be conveniently classified based on the effort and flow variables, material parameters, number of input and output ports, system structure topologies and input-output relationships.

Shown in Table 3 is the case representation for a solenoid and Figure 2 is a schematic of the solenoid with its systemic model, *i.e.* the effort and flow variables of its input and output ports. The solenoid converts electromagnetic energy into mechanical translational energy. The input port effort and flow variables are voltage, e_1 and current, f_1; the output effort and flow variables are mechanical force, e_2 and linear velocity, f_2. k is the solenoid constant and depends on the number of coil turns, the reluctance of the solenoid arm and the geometry of the configuration. Different solenoids with the same linear port relations differ in the value of k. In this bondgraph model of the solenoid we have not explicitly modelled the storage of energy in the magnetic core and the time delay it may cause in the movement of the arm. In fact the bond graph model provides the ability to model a single device at increasing levels of detail accounting for numerous physical effects. Shown in Table 4 is the case representation for a piston cylinder pump and Figure 3 shows the schematic and system model. The piston pump converts mechanical translational energy into hydraulic fluid energy. The input port effort and flow variables are mechanical force, e_1, and linear velocity, f_1; the output effort and flow variables are pressure, e_2, and flow rate, f_2. g is the pump constant and depends on the cross-sectional area of the cylinder and the piston. Different piston-cylinder mechanisms with similar linear port relations differ in the value of g.

The bond graph representation facilitates a principled way of combining simpler subsystems into more complex systems. Given the design specification of generating a pe-

Device name	Solenoid
Input port	`P1`
Output port	`P2`
Input port energy domain	`Electro-Magnetic`
Output port energy domain	`Mechanical-Translation`
Device relation	$e_2 = k.f_1, f_2 = e_1/k$
Components	`NIL`
Device topology	`NIL`

Table 3. Schema for a Solenoid

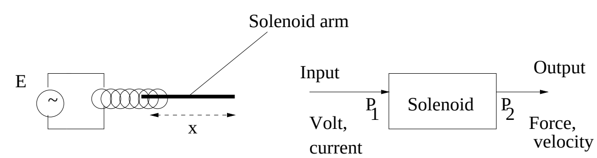

Figure 2. Schematic for a solenoid

Device name	Piston-cylinder (Pcyl)
Input port	`P3`
Output port	`P4`
Input port energy domain	`Mechanical-Translation`
Output port energy domain	`Hydraulic`
Device relation	$e_4 = g.e_3, f_4 = f_3/g$
Components	`NIL`
Device topology	`NIL`

Table 4. Schema for a Piston cylinder

No.	Relations	Example devices
1	$e_2 = k.e_1, f_2 = f_1/k$	Spur gear, Pipe
2	$e_2 = k.f_1, f_2 = e_1/k$	DC-motor
3	$e_2 = k.e_1.Sin(\mathcal{F}_1), f_2 = f_1/k.Sin(\mathcal{F}_1)$	Slider-crank mechanism
4	$e_2 = k.f_1.Sin(\mathcal{F}_2), f_2 = e_1/k.Sin(\mathcal{F}_2)$	Induction motor
5	$e_2 = k.e_1.\mathcal{G}(\mathcal{F}_1), f_2 = f_1/k.\mathcal{G}(\mathcal{F}_1)$	multi-bar linkages
6	$e_2 = \mathcal{H}(e_1), f_2 = f_1.e_1/\mathcal{H}(e_1)$	pipe networks

Table 2. Two port device relations

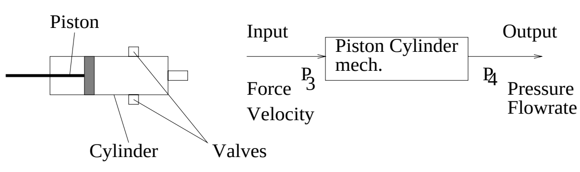

Figure 3. Schematic for a piston-cylinder

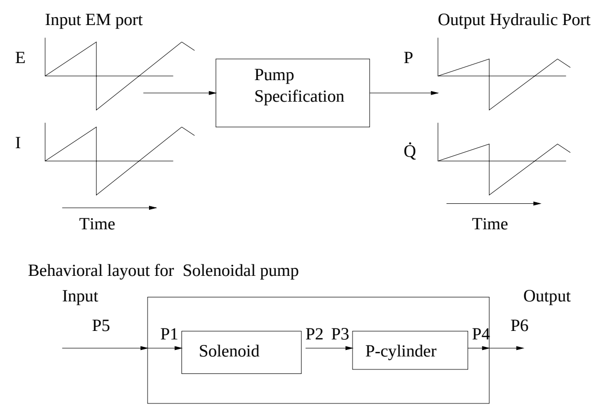

Figure 4. Schematic for a solenoidal pump

riodic pressure pulse to drive a fluid the two foregoing components can be combined to build a more complex electro-magnetic reciprocating pump as shown in Figure 4. The design specification for the pump shows the time history of the input and output effort and flow variables and also designates the energy domains of the input and output ports. The pseudo-code for case-based synthesis of two-port power devices is shown in Table 5. This code only provides the skeletal framework for the synthesis algorithm. Synthesis is guided by use of heuristics that use domain knowledge based on the relationships between ports, the topologies etc.

Following the steps in Table 5, we generate the combination of a solenoid and the piston-cylinder mechanism when the mechanical-translation power port is proposed as an intermediary port. Primary indices for retrieving cases are the energy domains of the ports, the relations, and inter-connective topologies. The two devices are combined at their ports as shown in the system model of Figure 4. The input-output relation for the pump is a combination of the device relations for the two components. Electrical power is converted into fluid energy. The case representation for the new device is shown in Table 6. The device relation for the solenoidal pump is obtained by *eliminating the effort and flow variables* at the shared common mechanical translation port from the solenoid and piston-cylinder system relations. In the device relation, n is a function of the solenoidal and piston-cylinder constants g and k. Further the system consists of two components connected in a cascade (serial) topology. The above pump mechanism is used to dispense regulated amounts of drugs in drug delivery systems. The electro-magnetic pump can pump varying amounts of fluid on varying the input current into the system.

Consider a new design specification that requires a mechanical pump to transfer fuel from a storage cell to a dispensing unit. A simple solution is to use the piston-cylinder system and combine it with a manually powered slider-crank mechanism as shown in Figure 5. The relationship for the slider crank mechanism was shown in Table 2. The piston cylinder component is *snippeted* out from the previous pump system since it generates fluidic power from linear mechanical motion and the system recognizes that a slider crank motion provides conversion of rotary power to translatory mechanical power. A combination of these two systems is guaranteed at the behavior level to provide the requisite power transformation.

Since our cases are indexed by behavior, the synthesis process also generates various physically different alternatives that exist in the case-base. Thus numerous mechanisms equivalent to the slider-crank mechanism are generated such as the scotch-yoke mechanism, eccentric-cam

1. Search the case-base for devices that provide power transformation from input energy domain to output energy domain. If any cases are retrieved, simulate for output, given the input behavioral specification and compare with output behavior specification. If failure, go to 2 else stop.
2. Propose an intermediate energy port in any of the five energy media. Search the case-base for all devices that provide power transformation from input energy domain to intermediate power port domain and from intermediate port to output port. Using the relations of each case in the pairs of cases retrieved, obtain relation between input and output ports by eliminating the intermediate port effort and flow variables. Simulate using the relation obtained and compare with output behavior specification. If failure, go to 3 else stop.
3. Repeat 2 by proposing additional intermediary ports in series. Stop when a predefined number of systems in series is reached.

Table 5. Pseudocode for synthesis of cases

Device name	Solenoidal pump
Input port	P5
Output port	P6
Input port energy domain	Electro-Magnetic
Output port energy domain	Hydraulic
Device relation	$e_6 = n.f_5, f_6 = e_5/n$
Components	Solenoid,Piston-cylinder
Device topology	P5-to-Solenoid-P1, Solenoid-P2-to-Pcyl-P3,Pcyl-P4-to-P6

Table 6. Schema for a solenoidal pump

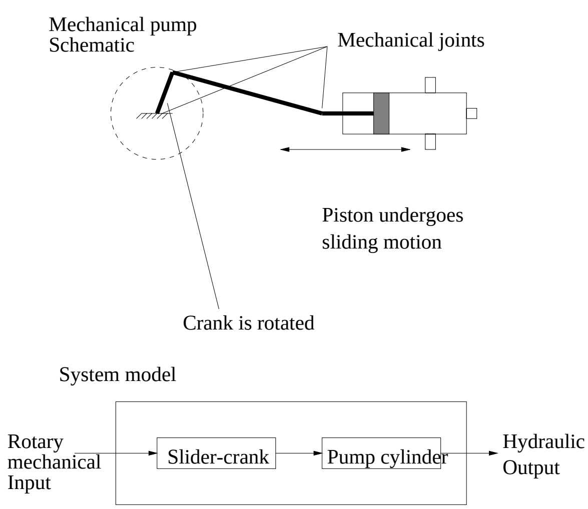

Figure 5. Schematic for a mechanical reciprocating pump

mechanism and a sinusoidal face-cam which are equivalent in the behavioral sense. In this manner, using behavioral indices for various physical realizations, we skirt the problem of shape synthesis but do provide useful physical and structural information to the designer. Further, as new physical realizations are made of components based on new manufacturing technologies and materials, these can be added to the case-base since their behavioral characteristics would still be encompassed by our device modeling framework.

The above examples illustrate the ability of the representation to facilitate composition of subsystems into more complex systems. Further, our implemented system can retrieve pieces of device components and reuse them in combination with other devices. These two characteristics of the representation, namely composability of subsystems and principled snippeting of subsystems from design cases, provide the backbone for developing design case bases. The representation also facilitates simulation of complex devices based on the input-output relations of the devices.

The discussion thus far has involved only two-port power devices in series topology. In the following section, we briefly describe the modeling and synthesis of other multiport power and signal devices.

5 EXTENSIBILITY OF THE REPRESENTATION

5.1 Multiple input-output and energy-storing devices

Networks of power systems can be generated using *three-port devices* as *distribution and confluence* elements. Multiple-input, multiple output power system topologies can be generated in a procedural manner by choosing various combinations of multiport systems in serial and parallel topologies.

Another characteristic of electro-mechanical systems is that they *store energy*. For example, in the electro-magnetic pump the solenoidal arm might be moved at a rapid pace. Motion of the arm may be controlled by adding an energy storing one-port element such as a spring. The spring stores some of the electro-magnetic energy during forward motion and provides damping to the solenoidal arm. Energy storage elements can be incorporated in a given device topology by choosing from a variety of one-port energy systems such as inductors and capacitances for electrical systems; flywheels and rigid bodies for mechanical systems.

5.2 Signal devices

Another class of devices that play a role in complex systems are devices that transfer *information* or signals. Signal devices are those devices that consume negligible amounts of energy but enable the transmission of signals from the input port to the output port. The signal devices establish relationships between physical variables at their respective input and output ports.

For example, an electrical fluid pressure transducer is a signal device. When placed in a pipe-line circuit, it consumes negligible power from the fluid flowing in the pipeline. Low power levels are sufficient to generate electrical readouts. Further, changes at the output end of the transducer do not affect the fluid flow on a large scale. In contrast, a hydraulic motor, in the circuit, driven by the fluid is a power device. If the power required to drive the load on the motor is high, the motor may stall, blocking fluid flow in the pipeline with a consequent drop in pressure and overall fluid energy.

The above example illustrates that signal devices are directional in nature allowing signal flow from input to output in contrast to power flow devices that allow for back-effects wherein changing the output can change the input. Signal devices provide for *unilateral* signal flow from input to output ports while power devices allow for *bilateral* signal flow, from input to output ports and back from output to input ports. *Signal devices are modelled as mathematical relationships between effort or flow variables at input and output ports without the constraint of energy conservation.* This allows one to model signal devices with both analog and digital relationships between the port variables. Further, common relations allow the system to retrieve devices with similar behavior from multiple energy domains. Thus a mechanical clutch and an electro-mechanical relay are similar in nature; both interrupt or allow power flow albeit the power is of two different kinds. Signal devices are used to monitor systems, provide information from the environment to the system and also change the behavior of a system by affecting its physical embodiment.

5.3 Linking parametric design

The relationships listed in the earlier tables (tables 2, 3, 4 and 6) have algebraic coefficients such as k and g which depend on the geometry and material properties of the device. These physical parameters can be fixed *apriori* as a design choice or varied during the functioning of a system. This feature provides another degree of freedom to the designer to obtain requisite input-output behavior. For example, in the electro-magnetic pump, a different pumping characteristic can be obtained by using a non-circular cross-section for the cylinder, an *apriori* choice, or by having a solenoid with a variable reluctance.

Our case-base consists of devices with values specified for these physical parameters. If the design specifications provide values and ranges for the effort and flow variables at the input and output ports, parametric design can be performed by searching for cases with satisfactory combination of values. For more details on bridging the conceptual and parametric design phases, see [15].

5.4 Linking configuration design

As mentioned in Section 2, configuration design is of concern after the physical realizations of different components are obtained and it is necessary to assemble them into one single unit. In our approach, we do not address the issue of modifying the form of these components, only ensuring that different components satisfy geometric constraints and form a viable assembly. Port variables in our bondgraph models may be treated as vectors and each behavioral port is mapped onto a port on the physically realized device. Each physical port in the has a specific location and orientation. Thus to ensure assembly, adjacent components may have to be geometrically compatible. The mapping between the behavioral port model and physical port model is one-to-many and our choice of components is restricted to those available in the case-base. Our current work is aimed at addressing how this assembly may be generated and extending this simple behavior port to physical port mapping model to encompass additonal geometric information.

6 RELATED DEVICE MODELS IN THE LITERATURE

In this Section, we present related approaches to modeling device behavior for purposes of behavioral synthesis of devices. Representations proposed in literature have tried to provide a qualitative abstraction and interpretation for the mathematical relations of physical effects. Devices have been primarily modelled as directional signal devices and the bilateral nature of power transfer in devices has been disregarded. Equations that model physical mechanisms have been represented as directed graphs with an *apriori* choice of directionality in [16, 17]. Further, these relations have been modelled in a qualitative manner in [17, 18]. Device models to represent only kinematics of linked mechanisms are presented in [19, 20, 21]. In [19], only linkage mechanisms are modelled in terms of configuration spaces. The focus is only on kinematics *i.e.* the relationship between flow variables at the input and output ports of mechanisms. There is no representation of dynamics nor other physical effects. In [20] a qualitative representation for mechanism motion is proposed. A predicate-logic formalism to model form dependence of mechanisms is proposed in [21], but is limited by its ability to deal with dynamic variations. In [6, 22], devices are modelled in terms of the Structure-Behavior-Function framework. The representation assumes that there exists a well-defined directional, mechanistic explanation for all physical mechanisms.

All the above mentioned approaches suffer from ambiguity, when multiple physical systems are combined in a single device. The notion of connectivity between subsystems ranges from sharing variables in graphs to predefined predicates. The rationale behind choosing and assembling subsystems is rather *ad hoc* and varies from method to method. Further, incomplete device models have inhibited validation of the devices synthesized in a consistent manner. Finally, scalability issues have not been addressed.

In our framework, power and signals flow through subsystems from input to output through *ports* where devices are connected. The relationships between port variables model the nature of the power and signal transformations i.e $\mathcal{S}$. Devices are composed by connecting subsystems at these ports and the process of synthesis is guided by the knowledge of the power transforming relations. Further, the structure of the device is provided in terms of the interconnective topology of the subsystems. Variations of the effort and flow variables at the input and output ports model the *observable* dynamic behavior of devices. The scalability of our proposed representation can be gauged by the success and use of bond graph models for behavioral analysis of a variety of electro-mechanical devices.

The bond graph formalism provides a comprehensive framework for understanding device behavior. It enables representation of device behavior in terms of the behavior of the components and also in terms of the behavior that emerges due to the interconnective topology of subsystems. However, there are limitations to this representation. It does not enable shape synthesis of components. A possible approach for synthesis of device geometry based on energy transfer between subsystems is suggested in [23]. The suggested approach starts with the principle of continuity of energy and the Poynting vector of energy flux and attempts to the solve the overall PDE with necessary boundary conditions so as to capture shape information. The continuity energy equation of interest is

$$\int_{\delta V} \mathbf{P}.\mathbf{n} d(\delta V) = \int_V \frac{\partial \epsilon}{\partial t} dV + \int_V g dV$$

The integral on the left describes the total energy flux through the system boundary; $\mathbf{P}$ is the generalized energy flux vector which gives the rate of energy transported per unit area, $\mathbf{n}$ is the normal to the system boundary of δV. On the right, the first term is the energy stored in the system and $\frac{\partial \epsilon}{\partial t}$ is the rate change in the energy density of the system and g is the rate of energy loss of the system. Essential problems here are to identify the nature of the various boundary conditions and constraints and develop efficient computational schemes. Bond graphs also start from the same energy continuity formulation and lump the spatial variables to generate a system of ODEs. It seems a compromise approach wherein one intelligently lumps physical form and otherwise solves a distributed formulation may prove to be a beneficial approach. Further research is required to model the detailed interaction between shape and behavior in a comprehensive manner.

7 CONCLUDING REMARKS

Bond graphs provide a consistent representation scheme to link various attributes considered in the design of a product.Though bond graphs have been primarily considered as a scheme for analyzing behavior, the bond graph language is comprehensive to support behavioral, parametric and configurational design and understand their various interactions. In a case-based design approach using bond graph based device representations, cases of designs can be indexed based on behavior, composed into complex systems, decomposed into subsystems and subsystems thus created can be reused in other systems. The representation facilitates the formulation of the behavioral synthesis task as a combinatorial search problem. Case-based synthesis of hybrid power and signal systems in a principled manner is facilitated by generating topologies of devices connected at their respective input and output ports. We have developed a representation for behavioral design specifications in terms of the time-histories of effort and flow variables. Further a computational scheme that allows for retrieving and combining cases based on specifications in a principled manner has been developed. At present, our implemented case base contains design cases from multiple energy domains. Our synthesis formalism allows use of cases from multiple energy domains to be reused and combined in a principled manner. Our current work investigates the development and use of domain dependent heuristics to guide this design task and integration of parametric and configurational design in our framework.

ACKNOWLEDGMENT

This research is sponsored by the Office of Naval Research, U.S. under contract no. ONR N00014-92-J-1298.

REFERENCES

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

[2] D. Karnopp and R. Rosenberg. *Analysis and Simulation of Multiport Systems.* The M.I.T. Press, 1968.

[3] J. R. Rinderle Finger, S. A transformational approach to mechanical design using a bond graph grammer. In *Proceedings of the First ASME International Design Theory and Methodology Conference*, September 1989.

[4] K. Ulrich and W. Seering. Computation and conceptual design. *Robotics and Computer-Integrated Manufacturing*, 4(3/4):309–315, 1988.

[5] R.C. Rosenberg. The bond graph as a unified data base for engineering system design. *ASME Journal of Engineering for Industry*, pages 1333–1337, November 1975.

[6] A. Goel. *Integration of case-based reasoning and model-based reasoning for adaptive design problem solving.* PhD thesis, Dept. of Computer and Information Science, Ohio State University, 1989.

[7] D. Navin Chandra, Sycara, K. Index transformation techniques for facilitating creative use of multiple cases. In *Proceedings of the twelfth International Joint Conference on Artificial Intelligence, (IJCAI-91)*, 1991.

[8] S. Narasimhan, Navin Chandra D., K.P. Sycara. Behavioral synthesis in cadet, a case-based design tool. In *Proceedings of the Seventh Conference on Artificial Intelligence Applications*, Miami, Florida, Feb, 1991. IEEE.

[9] G. Pahl and W. Beitz. *Engineering Design.* The Design Council, Springer-Verlag, London, 1984.

[10] V. Hubka. *Principles of Engineering Design.* Butterworth Scientific, London, 1980.

[11] W. Gosling. *The design of engineering systems.* Wiley, 1962.

[12] Navin Chandra, D, K. Sycara and S. Narasimhan A transformational approach to case based synthesis. *AI EDAM*, 5(1), 1991.

[13] D. Karnopp and R. Rosenberg. *System Dynamics: A Unified Approach.* John Wiley Sons, New York, 1975.

[14] R. Rosenberg and Karnopp. D. *Introduction to Physical System Dynamics.* McGraw-Hill Book Co., 1983.

[15] T.N.Madhusudan. A bond graph based approach to case–based synthesis. Technical Report CMU-RI-TR-95-29, The Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA 15213, July 1995.

[16] B. Williams. *Invention from First Principles via Topologies of Interaction.* PhD thesis, Massachusetts Institute of Technology, 1989.

[17] D.Navin-Chandra *et al.* Cadet: A case-based synthesis tool for engineering design. *International Journal of Expert Systems*, 4(2):157–188, 1992.

[18] J. DeKleer and Brown.J. A qualitative physics based on confluences. *Artificial Intelligence*, 24:7–83, 1984.

[19] S. Addanki Joskowicz, L. From kinematics to shape: An approach to innovative design. In *Proceedings of the seventh National Conference on Artificial Intelligence (AAAI88)*, pages 347–352, 1988.

[20] S. Kota. A qualitative matrix representation scheme for the conceptual design of mechanisms. In *Proc. of the ASME Design Automation Conference (21st Biannual ASME Mechanisms Conference)*, 1990.

[21] Srikanth M. Kannapan and Kurt M. Marshek. An algebraic and predicate logic approach to representation and reasoning in machine design. *Mechanisms and Machine Theory*, 25(3):335–353, 1990. To be published.Locate journal issue and volume.

[22] A. Goel and B. Chandrasekaran. Integrating model-based reasoning and case based reasoning for design problem solving. In *AAAI-88 Design Workshop*, St. Paul, MN., Expected in 1989.

[23] Vadim Shapiro and Herb Voelcker. On the role of geometry in mechanical design. *Research in Engineering Design*, pages 69–73, 1989.