Adaptive Modeling

Ashok K. Goel

College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
goel@cc.gatech.edu

Abstract

We describe a computational technique for functional modeling of physical devices. In this technique, functions and behaviors of a given device are derived by retrieving the structure-behavior-function (SBF) model of a structurally similar device and revising the retrieved model to meet the specifications of the given device. The SBF model of a device explicitly represents its structure, its functions, and its internal causal behaviors that specify how its structure delivers its functions. The model of the known device is revised by model-revision plans, where each plan accommodates a specific type of structural difference between the new and the known devices. The model ontology gives rise to a classification of structural differences and corresponding model-revision plans. The process of model revision is focused by the organization of the known model. The revised model for the new device is stored in memory for potential reuse in future. We call this computational process *adaptive modeling.*

Motivations, Background, and Goals

Functional models of devices have proved to be quite useful for reasoning about a variety of function-related tasks. The Functional Representation (FR) scheme [Sembugamoorthy and Chandrasekaran 1986; Chandrasekaran, Goel and Iwasaki 1993], for example, has been extensively used for diagnosis (e.g., [Sticklen and Chandrasekaran 1989]), redesign (e.g., [Goel and Chandrasekaran 1989]), and, more recently, design verification (e.g., [Iwasaki and Chandrasekaran 1992)]. Since functional models explicitly represent the functions of a device and use the function representations to organize behavioral knowledge about the device, they help define problem spaces for function-related tasks, and provide access to the knowledge relevant for searching the spaces.

But the origin, generation and acquisition of functional device models remain open issues. Not only are these questions fundamental, but, in addition, their answers are likely to impose additional representational constraints on the models. The research described here is motivated by both goals: exploration of the origin, generation and acquisition of functional device models, and discovery of additional representational constraints on the models.

The origin of this research lies in our earlier work on the conceptual phase of functional device design. The tasks of conceptual (or preliminary) device design and (qualitative) device modeling have an "inverse relationship" with each other. The task of device design takes as input a specification of the desired output behaviors of the device, i.e., the device functions. It has the goal of giving as output a specification of the structure of the device that can deliver the desired functions. Thus, the task of device design is a *function → structure* mapping. The task of device modeling takes as input a specification of the structure of a device. It has the goal of giving as output a specification of the output behaviors of the system. Thus, the task of qualitative device modeling is a *structure → output behavior* mapping. Since device functions are a subset of the output behaviors of the device, it follows that the two tasks, though not exact inverses of each other, have an inverse relationship[1].

In our work on the conceptual phase of functional design, this inverse relationship between device design and device modeling led to us to hypothesize that knowledge of device models may facilitate the adaptation of the designs of known devices to design new devices of similar functionality. If the *structure → output behavior* map of a device was known, then, we hypothesized, this map may enable adaptation of the device structure for achieving a different, though similar and related, set of device functions. In the Kritik family of systems, we have extensively investigated this hypothesis [Goel 1991, 1992; Goel and Chandrasekaran 1992]. Kritik contains a design case memory, where each case in the memory specifies a structure-behavior-function (SBF) model that explains how the structure of the de-

[1] The "output behaviors" of a device, in our terminology, include both the intended and the unintended behaviors. The "functions" of the device refer to the intended output behaviors. The "internal behaviors" of the device, in this terminology, are the causal processes that result in its output behaviors including the device functions.

vice delivers its functions[2]. The ontology of the SBF models gives rise to a classification of functional differences between related devices. Kritik contains skeletal plans for design modification, where each skeletal plan can help to reduce a given type of functional difference. When the function of desired device is specified, Kritik retrieves a functionally similar design from its case memory along with its SBF model. Also, it notes the differences between function desired of and delivered by the retrieved design, and instantiates corresponding design modification plans. The instantiation of the skeletal plans in the context of the SBF model of the retrieved design leads Kritik to a modified design along with its (revised) SBF model. The model of the modified design enables design verification, and the verified design is stored in the design case memory for potential reuse. In this way, Kritik uses qualitative models of known devices to generate candidate designs. We call this computational process *adaptive design*.

But note that the above process of adaptive design also generates SBF models for the new device designs by revising the SBF models of the known devices. In analogy to the technique of adaptive design, this has led us to think in terms of an adaptive technique for the general task of qualitative modeling of physical devices. In this technique, given the structural specification of a device, some of the output behaviors of a device are derived by locally revising the (SBF) model of a structurally similar device. The model is revised through instantiation and execution of skeletal model-revision plans, where each plan accommodates a specific type of structural difference between the new and the known devices. The ontology of SBF models gives rise to a classification of structural differences and model-revision plans. The process of model revision is focused by the organization of the internal causal behaviors of the SBF model of the known device. The revised SBF model for the new device is stored in a model memory for potential reuse in future. We call this computational process *adaptive modeling.*

The goal of this paper is to describe the technique of adaptive modeling, We illustrate the technique using a simple example that originates from the Kritik system. But while Kritik used the technique of adaptive design to perform *function → structure* mappings, the new technique of adaptive modeling performs the inverse *structure → function* mappings. In the following discussion of adaptive modeling, we focus on the issues of the content and representation of SBF models of known devices, and the computational process of revising SBF models to derive the functions of new, but similar and closely related, devices.

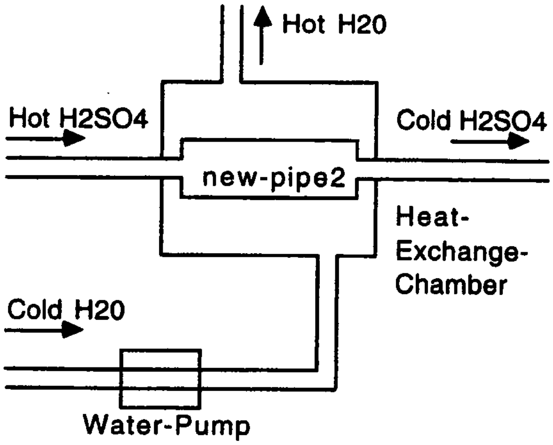

Figure 1: The Sulfuric Acid Cooler

An Illustrative Example

As an illustrative example, let us consider the task of qualitatively modeling the Sulfuric Acid Cooler (SAC) schematically depicted in Figure 1. We are given a specification of the structure of SAC, where the structural specification includes the specification of the components comprising the system, and the structural and behavioral interactions among them. It includes, for example, the specification that the Sulfuric Acid in SAC is highly acidic, and the pipes through which Sulfuric Acid flows are made of a material that allows only high-acidity liquids.

Let us suppose that a model memory contains a number of device designs along with their qualitative device models. The technique of adaptive modeling, in general, sets up four subtasks of the qualitative-modeling task: (i) *model retrieval*, (ii) *model revision*, (iii) *model evaluation*, and (iv) *model storage.* Given the structural specification of a device, such as SAC, the first subtask retrieves the design of a structurally similar device along with its qualitative model. If the structure of the retrieved design exactly matches the structure of the specified device, then the model of the retrieved design directly specifies the functions of the device, and the processing is terminated. Else, the second subtask revises the model of the retrieved design to meet the structural specification of the given design. The third subtask evaluates the revised model for internal consistency. If the model-evaluation task finds that the revised model is internally inconsistent, then the model can be revised again, or alternatively, another model can be retrieved from memory and revised to produce the desired model. The fourth and final task stores the evaluated model in the design model memory for potential reuse in future.

[2] The SBF device models are closely related to, but distinct from, the FR scheme. The 'B' in SBF models stands for the internal behaviors of the device that explain how the behaviors of the structural elements of the device get composed into the device functions.

Structure-Behavior-Function Models

Let us assume that the design model memory contains the design of a Nitric Acid Cooler (NAC) along with its SBF model. Let us also assume that the Nitric Acid in NAC has a low acidic content, and the pipes through which Nitric Acid flows in NAC allow only low-acidity liquids. In this section, we briefly describe the content and representation of SBF device models using NAC as an illustrative example.

The SBF model of a device explicitly specifies (i) its structure, (ii) its functions (i.e., the intended output behaviors of the device), and (iii) its internal causal behaviors. The causal behaviors explain the functional role of each structural component of the device, and the causal processes that compose the functions of the structural components into the device functions.

Structure

The structure of a device is viewed as constituted of components, substances, and relations among them. The structure of SAC shown schematically in Figure 1, for example, is viewed as composed of Sulfuric Acid and Water, a water-pump, a heat-exchange chamber, and various pipes, where the pipes through which Sulfuric Acid flows are connected in series, etc. The substances can be abstract, e.g., heat, electrical charge, angular momentum. Substances can flow from one component to another if (and only if) the two components are connected to each other. SBF device models thus are flow models.

The structural model in a SBF device model is represented as a schemata that specifies not only the components and substances constituting the structure of the device but also. the structural and behavioral interactions among them. It specifies, for example, that pipe2 (in NAC) is *included* in the heat-exchange-chamber, and *allows* the flow of a low-acidity substance from one end to another. The SBF ontology borrows the taxonomy of primitive device behaviors from Bylander 's [1991] work on composing the behaviors of the structural elements of a device into the behaviors of the device. The specification of a structural component includes its functional abstractions in terms of the primitive device behaviors. The component specification also contains pointers to the causal behaviors in which it plays a role.

Function

Device functions are viewed as transformations from an input behavioral state to an output behavioral state. A function is represented in the form of schema. The schema for a function specifies the behavioral state it takes as input and the behavioral state it gives as output. It also specifies the internal causal behaviors responsible for the achievement of the function; thus the functions of a device act as indices to its internal behaviors. In addition, the function schema specifies the conditions under which the internal behavior accomplishes the function including the stimulus from the environment which triggers the internal behavior. Function representation in SBF models builds on Chandrasekaran's FR scheme [Sembugamoorthy and Chandrasekaran 1986; Chandrasekaran, Goel and Iwasaki 1993]. Unlike FR, however, the SBF ontology gives rise to a vocabulary for expressing the semantics of behavioral states.

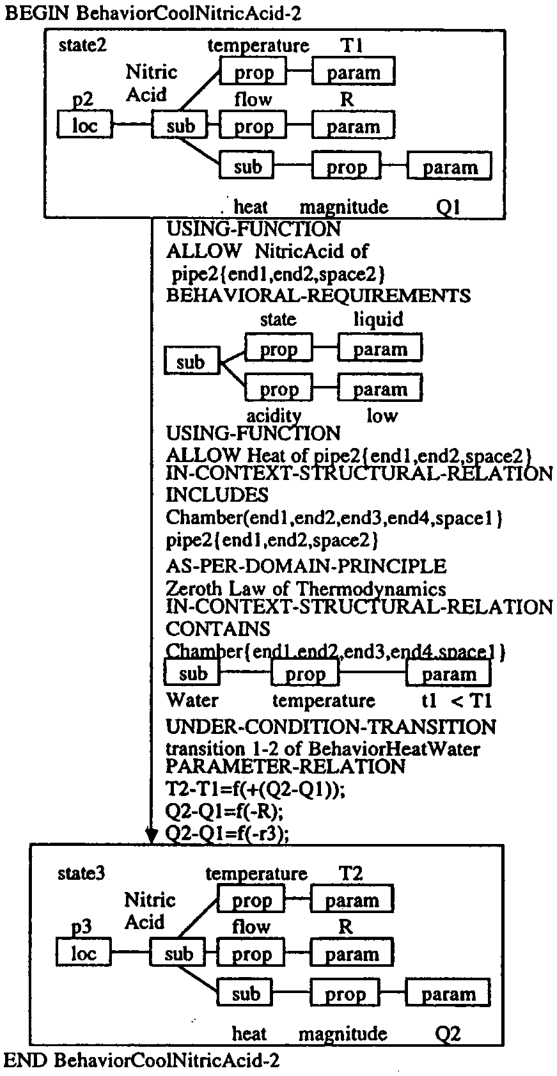

Figure 2: A Fragment of an Internal Behavior of the Nitric Acid Cooler

Behavior

Finally, the SBF model specifies the internal behaviors of the device. The internal behaviors are causal processes that compose the structural and behavioral interactions between the components and substances into the functions of the device. A small fragment of one internal behavior of NAC is shown in Figure 2.

Knowledge of an internal behavior is represented as a directed acyclic graph (DAG). The nodes in a DAG represent the behavioral states of the device and the links represent the state transitions. A node (i.e., a behavioral state) is represented in the form of component and substance schemas. For example, the substance schema representing *state*2 in Figure 2 specifies the location, property, and parameter of a substance, viz., it specifies that the location of substance *Nitric Acid* is p_2 (where p_2 is a point in the device space), and the parameter of property temperature is T_1. It also specifies that the substance *Nitric Acid* contains another substance, *heat*, which has its own properties and parameter values.

A state transition in an internal behavior is also represented as a schema. The slots in the transition schema act as annotations on the state transitions, and specify the causes of the transitions. For example, the transition *state*2 ⇒ *state*3 in BehaviorCoolNitricAcid-2 specifies that one of the causes of the transition is the function *allow* low-acidity liquids of pipe2. A state transition may be annotated by the function of some component (e.g., USING-FUNCTION ALLOW), enabling conditions such as a specific structural relation between some components (e.g., INCLUDES Chamber Pipe2), a domain principle (e.g., Zeroth Law of Thermodynamics), and qualitative parametric equations (e.g., $T_2 - T_1 = f(+(Q_2 - Q_1)))$ as shown in Figure 2. Since the device functions act as indices into the internal behaviors responsible for their accomplishment, and since a state transition in an internal behaviors may index the function of a device component, this leads to hierarchical organization of the device model of the form *function* ⇒ *behavior* ⇒ *function* ⇒ *behavior* ... Again, SBF models borrows this organizational scheme from Chandrasekaran's FR scheme.

Within a given level of this hierarchy, the internal behaviors are organized along the flow of specific substances. This organization builds on Govindaraj's [1987] work on qualitative approximations of quantitative device models. Interactions between different internal behaviors are specified through enabling and disabling pointers. For example, the transition *state*2 ⇒ *state*3 in BehaviorCoolNitricAcid-2 specifies UNDER-CONDITION-TRANSITION *transition*1 − 2 of BehaviorHeatWater as an enabling condition as shown in Figure 2. Thus, one internal behavior may index another internal behavior.

Model Revision Process

In this section, we briefly describe the functional architecture and computational process of model revision in adaptive modeling. The SBF model of a known device (such as NAC) is revised by a family of skeletal model-revision plans. The functional architecture of the model revisor consists of (i) a *plan memory*, (ii) a *plan selector*, (iii) a *plan executor*, and (iv) *a difference orderer*. The SBF language described in the previous section leads to a typology of structural differences between two devices. The plan memory contains a plan for each type of structural difference. The stored plans are indexed by the types of structural differences to which they are applicable. Given a specific type of structural difference between the new and the known devices, the plan selector retrieves the applicable model-revision plan from the plan memory. The plan executor instantiates the retrieved plan in the context of the structure-behavior model of the known device, and executes it on the model to produce a structure-behavior model for the new device. If the new and the known devices differ in more than way, then the difference orderer heuristically ranks the structural differences by the order of the difficulty of accommodating them, and the plans applicable to more difficult structural differences are retrieved and executed before the plans applicable to less difficult ones.

A model-revision plan specifies a compiled sequence of abstract operations. Different plans specify different sequences of (possibly) different operations. The abstract operations in the plans are specified in the vocabulary of SBF language such as those enclosed in boxes in Figure 2 (e.g., *sub*, *prop*), or shown in capital letters (e.g., USING-FUNCTION, ALLOW).

To illustrate the model-revision process, let us return to the example of modeling the Sulfuric Acid Cooler (SAC) shown in Figure 1. Recall that the model-retrieval task resulted in the selection of the model for the Nitric Acid Cooler (NAC), where the structures of SAC and NAC differ in that (i) while SAC contains high-acidity Sulfuric Acid, NAC contains low-acidity Nitric Acid, and (ii) while the pipes through which Sulfuric Acid flows in SAC allow high-acidity liquids, the pipes through Nitric Acid flows in NAC allow only low-acidity liquids. The first of these two differences, *Nitric Acid* → *Sulfuric Acid*, is an instance of the *substance substitution* type of structural differences; the second, *pipe (allow low-acidity substances)* → *pipe (allow high-acidity substances)*, is an instance of the *component replacement* type of structural differences. Since the structures of SAC and NAC differ in more than way, and, since, in the class of domains of interest, revising a component-substance model to accommodate the structural difference of component replacement is in general more difficult than revising it to accommodate the difference of substance substitution, the difference orderer ranks the two differences between the structures of SAC and NAC so that the model for NAC is first revised for the difference of *pipe (allow low-acidity substances)* → *pipe (allow high-acidity substances)*, and then for the difference of *Nitric Acid* → *Sulfuric Acid*.

Let us consider the revision of the model for NAC, given the structure difference of *pipe (allow low-acidity substances)* → *pipe (allow high-acidity substances)* between the structures of NAC and SAC. This structural difference is used as probe into the plan memory to access the model-revision plan for *component-replacement*. The model-revision plan for component-replacement revises the component-substance model for NAC in several steps. First, from the specification of the old pipe (pipe2) in NAC, it determines that the pipe2 plays a functional role in BehaviorCoolNitricAcid-2. Then, it decomposes BehaviorCoolNitricAcid into three segments: (i) the transition in which the old component (pipe2) plays a functional role (in the present example, *state*2 ⇒ *state*3 in BehaviorCoolNitricAcid shown Figure 2, (ii) the sequence of state transitions preceding it (not shown here), and (iii) the sequence of state transitions succeeding it (also not shown here). Next, the transition in which the old component (pipe2) plays a role is revised by replacing the behavioral abstraction of pipe2 (which allows the flow of low-acidity liquids) by the behavioral abstraction of the new pipe (newpipe2) in SAC (which allows the flow the high-acidity liquids). Then, the revised transition is composed with the preceding and succeeding segments of the original behavior to obtain the revised behavior. Finally, the constraints introduced by the new component, represented by changes in the values of the variables characterizing the old and new components (pipe2 and newpipe2), are propagated forward through the newly composed behavior to obtain the revised internal behavior and function. Also, the schema for the function is revised by associating a pointer with it to the revised internal causal behavior.

Similarly, the structural difference *Nitric Acid* → *Sulfuric Acid* between the structures of NAC and SAC is used to access the model-revision plan for *substance substitution*. The model-revision plan for substance substitution further revises the (already revised) internal behaviors and functions in the model for NAC by replacing the old substance (low-acidity Nitric Acid) by the new substance (high-acidity Sulfuric Acid). This produces a SBF model for the Sulfuric Acid Cooler (SAC). The function schema in the SBF model specifies the function of SAC, and the internal behaviors specify in the model specify how the structure of SAC results in the function.

Discussion

In this section, first we describe current work on the evaluation of the technique of adaptive modeling, and then we relate the technique to other research on qualitative models and modeling.

Evaluation

In the Kritik series of systems, we have extensively evaluated the technique of adaptive design. We briefly summarize the main lessons from the evaluation of the adaptive-design technique because we expect that the technique of adaptive modeling is likely to be constrained by similar factors. Our analysis of the Kritik work shows that the technique of adaptive design is useful whenever (i) the differences between the functions of the desired and the known devices are small, (ii) design-modification plans for each specific type of functional difference between the desired and the known devices are available, (iii) detailed knowledge of the internal behaviors of the known devices also is available, including knowledge of the causal dependencies between the behavioral states and the functional role played each component and substance in the state transitions, and (iv) the structural modifications to the known design needed to reduce the functional differences can be localized to specific components and/or substances. For the class of design problems for which the above conditions are met, adaptive design provides a productive and feasible technique for the creation of new designs through model-based modification of old designs.

Our work on adaptive modeling is of more recent origin, and only now have we begun to systematically evaluate it. In one experiment, we have developed a knowledge system called KA [Goel et al 1996, Peterson, Mahesh and Goel 1994] that acquires qualitative device models from device descriptions in the popular science book "The Way Things Work" [Macaulay 1988]. Two characteristics of this book relevant to our discussion are that (i) each device description is accompanied by a cutaway diagram that reveals the device structure, and (ii) many devices are described through reference to other devices. For example, the very short text on the common fire extinguisher is accompanied by detailed diagrams of its structure and makes explicit reference to the common spray can. Macaulay's implicit assumption appears to be that given the structure of the fire extinguisher and a qualitative model of how the spray can works, the reader can easily build a similar qualitative model of the fire extinguisher. Accordingly, we supply KA with a complete and detailed SBF model of the spray can and a structural specification of the fire extinguisher. We have found that KA can use the technique of adaptive modeling to autonomously acquire a SBF model of the fire extinguisher by adapting the model of the spray can. This appears to indicate that the SBF language is adequate for representing both the fire extinguisher and the spray can, at least up to the coarse level of one popular science book. It also seems to indicate that the technique of adaptive modeling is computationally feasible and actually works for complex examples taken from a real book.

In another experiment, we are developing a knowledge system called Torque [Griffith et al 1996] that acquires SBF models of hypothetical devices by adapting SBF models of real devices. Torque seeks to model a verbal protocol collected by Clement [1989]. In this protocol, a scientist answers questions about how much an ordinary spring may stretch when an external force is applied to it. In the course of constructing answers to questions about the spring, the subject imagines a series of hypothetical devices similar to the ordinary spring, for example, a spring with only one coil, and a spring with square coils. He constructs qualitative models of the hypothetical devices by revising his model for the spring, and uses the constructed models to verify his answers to the questions concerning the original spring. Nersessian [1995] has argued that this kind of constructive modeling is a core process in scientific problem solving and theory formation. Torque uses the technique of adaptive modeling for constructing SBF models of the hypothetical devices by adapting the SBF model of the spring.

Related Research

We have already indicated the relationship between our SBF device models and Chandrasekaran's, Bylander 's and Govindaraj's work on device representation. In research in Cognitive Engineering, Rasmussen [1985] has proposed a hierarchical organization for presenting device knowledge to human users. Like our SBF models, his SBF device models too specify the structure, the behaviors, and the functions at each level in the hierarchy. In Design research, [Umeda et al 1990] have described similar FBS device models. Like in our SBF models, in their FBS models behaviors mediate between function and structure.

One common (and justified) criticism of functional device models has been that they are not generative. Another common (and, again, justified) criticism has been that the functional representations are underconstrained. Adaptive modeling is a technique for generating and acquiring one class of functional device models called SBF models. Our earlier work on adaptive design too resulted in the derivation of SBF models of new devices through revision of SBF models of known devices. Our current work on adaptive modeling extracts and generalizes this technique for generating SBF device models. As we continue the development of this technique, we expect it to lead to additional representational constraints on SBF models.

Bylander [1991] describes an alternative technique for deriving the output behaviors of a device. In his method, called *consolidation*, the behaviors of a given device are derived by composing the (primitive) behaviors of its structural components. The behavior composition is governed both by the structural relations among the components and by general rules of composition. The composition process is driven by iterative application of the compositional rules. This process results in a specification of potential output behaviors of the device.

The technique of adaptive modeling complements compositional methods such as consolidation. Compositional methods appear to be more general because they can derive all the potential output behaviors of the device. But they are computationally expensive, and may require post processing to determine the actual behaviors of the device from its potential behaviors. The adaptive method appears to be more restricted in its applicability. For example, the derivation of the output behaviors of SAC in our example is limited to the device functions. This is because our SBF model of NAC specifies only the functions of the device, not all of its output behaviors. Of course, we could enhance the SBF model of NAC by specifying more of its output behaviors. But the method still would be limited by the completeness of the enhanced SBF model.

Note that the technique of adaptive modeling too has a compositional aspect. The application of the model-revision plan of *component replacement*, for example, first deletes the functions of the old component from the internal causal behaviors of the SBF model, and then adds the functions of the new component to them. But this composition in adaptive modeling is localized both by the organization of the SBF model and by the model adaptation goals.

Our technique of adaptive modeling is also related to some work on theory revision by analogical transfer. Falkenhainer [1989], for example, has integrated the technique of *structure mapping* [Gentner 1983] with *qualitative process* representations [Forbus 1984] for completing almost complete theories of novel physical processes by analogy to complete theories of known processes. However, the generality and scalability of this kind of analogical transfer remains an open question. The technique of adaptive modeling is closer to case-based reasoning (e.g,, [Kolodner 1993]) which emphasizes almost complete analogical transfer between similar and related situations instead of highly-selective transfer between distant situations.

Acknowledgements

This work has been supported by the National Science Foundation (research grant IRI-92-10925), the Office of Naval Research (research contract N00014-92-J-1234), and Northern Telecom (research gift).

References

T. Bylander. A Theory of Consolidation for Reasoning about Devices. *Man-Machine Studies.* Vol. 35, pp. 467-489. 1991

B. Chandrasekaran, A. Goel, and Y. Iwasaki. Functional Representation as Design Rationale. *IEEE Computer*, 48-56. January 1993.

J. Clement. Learning via model construction and criticism. In *Handbook of Creativity: Assessment, Theory and Research*, G. Glover, R. Ronning and C. Reynolds (editors), pp. 341-381, New York: Plenum,

B. Falkenhainer. Learning from Physical Analogies: A Study in Analogy and the Explanation Process. Ph.D. Dissertation, Department of Computer Science, University of Illinois at Urbana; 1989.

K. Forbus. Qualitative Process Theory. *Artificial Intelligence*, 24:85-168, 1984.

D. Gentner. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7(2), 1983.

A. Goel. A Model-based Approach to Case Adaptation. *Proc. Thirteenth Annual Conference of the Cognitive Science Society*, August 1991, pp. 143-148. Lawrence Erlbaum Associates.

A. Goel. Representation of Design Functions in Experience-Based Design. In *Intelligent Computer Aided Design*, pp. 283-308. D. Brown, M. Waldron and H. Yoshikawa (editors). North-Holland. 1992.

A. Goel and B. Chandrasekaran. Functional Representation of Designs and Redesign Problem Solving. In *Proc. Eleventh International Joint Conference on Artificial Intelligence*, pp. 1388-1394. Morgan Kaufmann Publishers. 1989.

A. Goel and B. Chandrasekaran. Case-Based Design: A Task Analysis. In *Artificial Intelligence Approaches to Engineering Design, Volume II: Innovative Design*, pp. 165-184. Tong and D. Sriram (editors). Academic Press. 1992.

A. Goel, K. Mahesh, J. Peterson and K. Eiselt. Multistrategy Language Understanding for Device Comprehension. To appear in *Proc. 1996 Cognitive Science Conference*, San Diego, July 1996, Hillsdale, NJ: Erlbaum.

T. Govindaraj. Qualitative Approximation Methodology for Modeling and Simulation of Large Dynamic Systems: Applications to a Marine Power Plant. *IEEE Transactions on Systems, Man and Cybernetics.* Vol. SMC-17 No. 6, pp. 937-955. 1987.

T. Griffith, N. Nersessian and A. Goel. The Role of Generic Models in Conceptual Change. To appear in *Proc. 1996 Cognitive Science Conference*, San Diego, July 1996, Hillsdale, NJ: Erlbaum.

Y. Iwasaki and B. Chandrasekaran. Design Verification Through Function- and Behavior-Oriented Representation: Bridging the Gap Between Function and Behavior. *Proc. Second International Conference on AI in Design*, Pittsburg, July 1992, pp. 597-616, Boston: Kluwer.

J. Kolodner. *Case-Based Reasoning.* Morgan Kaufmann Publishers, San Mateo. 1993.

D. Macaulay. *The Way Things Work.* Boston: Houghton Mifflin Company, 1988.

N. Nersessian. Should Physicists Preach What They practice? Constructive Modeling in Doing and Learning Physics. *Science and Education*, 4:203-226, 1995.

Justin Peterson, Kavi Mahesh, Ashok Goel, and Kurt Eiselt. KA: Situating Natural Language Processing in Design Problem Solving. In *Proc. Sixteenth Annual Conference of the Cognitive Science Society*, August 1994, Atlanta, Georgia, pp. 711-716, Hillsdale, NJ: Lawrence Erlbaum.

J. Rasmussen. The Role of Hierarchical Knowledge Representation in Decision Making and System Management. *IEEE Trans. Systems, Man and Cybernetics*, 15:234-243, 1985.

V. Sembugamoorthy and B. Chandrasekaran. Functional Representation of Devices and Compilation of Diagnostic Problem Solving Systems. In *Experience, Memory and Reasoning*, Kolodner and and Riesbeck (editors), Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1986.

J. Sticklen and B. Chandrasekaran, et al. (1989). "Distributed Causal Reasoning." Knowledge Acquisition 1: 139-162.

Y. Umeda, H. Takeda, T. Tomiyama, and H. Yoshikawa. Function, Behavior and Structure. In *Proc. Fifth International Conference on Applications of AI in Engineering*, Vol. 1, pp:177-193, 1990.