Generic Teleological Mechanisms and their Use in Case Adaptation

Eleni Stroulia and Ashok K. Goel *
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
eleni@cc.gatech.edu, goel@cc.gatech.edu

Abstract

In experience-based (or case-based) reasoning, new problems are solved by retrieving and adapting the solutions to similar problems encountered in the past. An important issue in experience-based reasoning is to identify different types of knowledge and reasoning useful for different classes of case-adaptation tasks. In this paper, we examine a class of non-routine case-adaptation tasks that involve patterned insertions of new elements in old solutions. We describe a model-based method for solving this task in the context of the design of physical devices. The method uses knowledge of generic teleological mechanisms (GTMs) such as cascading. Old designs are adapted to meet new functional specifications by accessing and instantiating the appropriate GTM. The Kritik2 system evaluates the computational feasibility and sufficiency of this method for design adaptation.

Overview

In experience-based (or case-based) reasoning, new problems are solved by retrieving and adapting solutions of similar problems encountered in the past. Once a new solution is created, it can be stored in memory for potential reuse in future. Much of previous work on modeling experience-based reasoning uses simple modification operators and rules for "tweaking" the solution in the retrieved case (Alterman 1988; Ashley & Rissland 1988; Hammond 1989; Kolodner & Simpson 1989). These methods are often sufficient for routine case-adaptation where the needed modifications involve changing the parameter of an element in the old solution or substituting one solution element by a similar one. Many adaptation tasks, however, appear to require modifications that go beyond parameter changes or component substitutions. Case adaptation in innovative design, for example, often involves insertion of new components in old designs.

Reasoning about insertions of new components in a design structure can be very complex. This is because the insertion of a new component can potentially have a non-local impact on the functionality of the design. A general, computationally feasible and cognitively plausible model for this type of reasoning is not yet known. Nevertheless, it seems reasonable to assume that (human) designers use additional knowledge to constrain their reasoning about design modifications, and thus manage the complexity of the task. An important and open research issue in modeling experience-based reasoning, then, is to identify additional types of knowledge useful for different classes of case-adaptation tasks and to develop process models of their usage.

Informal observations of designers have led us to hypothesize that (i) one class of case-adaptation tasks is characterized by *insertion of specific patterns of components* into the structure of the design retrieved from the case memory, and (ii) the insertion of these patterns is based on knowledge of *generic teleological mechanisms*. Examples of generic teleological mechanisms (GTMs) in design include cascading, feedback, and feedforward. These mechanisms are "teleological" in that they result in specific functions. The mechanism of feedback, for example, takes information about the deviation of the output of a system from its desired output, feeds it back into an input to the system, and this results in the specific function of reducing the deviation in the output. Also, these mechanisms are "generic" in that they are device independent.

The feedback mechanism, for example, is independent of the specific device in which it might be instantiated, and in principle can be instantiated in any control system. The instantiation of such a mechanism in the context of a particular device leads to a patterned insertion of components in the structure of the system. The instantiation of the feedback mechanism in a system, for example, may result in the insertion of components that can measure the deviation of the system's output from its desired output, compo-

*This work has been supported by a research grant from the Office of Naval Research, contract N00014-92-J-1234, a graduate fellowship from IBM, a research gift from NCR, and equipment donated by IBM and Symbolics.

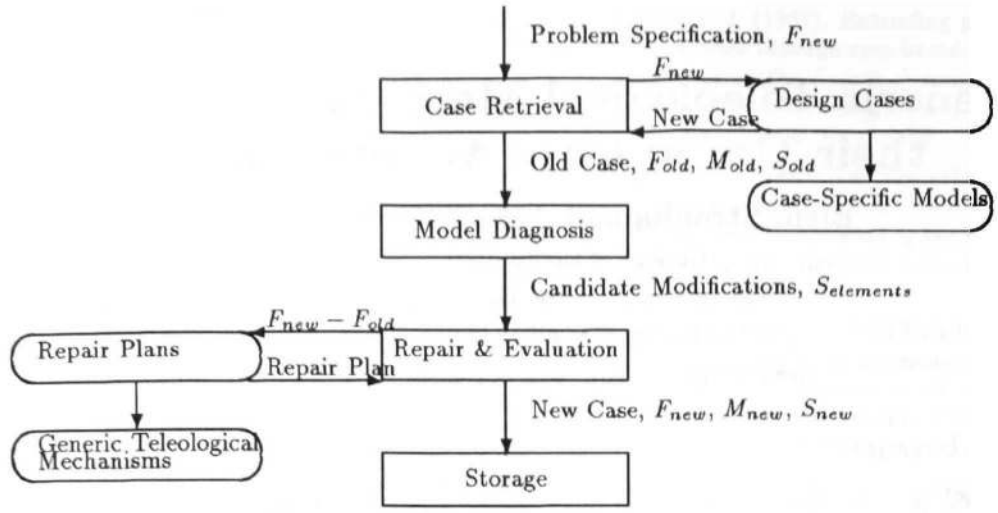

Figure 1: Kritik2's Process Model

nents that can relay this information to an input to the system, and components that can control the system input accordingly. This hypothesis about case adaptation raises a number of issues concerning the representation, indexing, access, and use of GTMs. The Kritik2 project investigates these issues in the context of designing physical devices such as simple electrical circuits and heat exchangers.

Process Model

Figure 1 depicts Kritik2's process model for experience-based design. Kritik2 takes as input the specification of a function desired of a new device F_{new}. In the case-retrieval step, it uses F_{new} as a probe into a functionally indexed case memory and retrieves the closest matching case. Each design case in the case memory contains a pointer to the corresponding device model M_{old} that specifies how the structure of the known device S_{old} delivers its functions F_{old}. In the diagnosis step, Kritik2 uses M_{old} to generate candidate modifications to S_{old} so as to achieve F_{new}. In the repair step, it uses the difference between F_{new} and F_{old} as a probe into a memory of repair plans and retrieves the applicable plans. The candidate modifications generated by the diagnosis step are used as a secondary index to discriminate among the applicable plans. Some repair plans contain pointers to GTMs. If selected, such a repair plan instantiates the corresponding GTM in the context of M_{old} and synthesizes the instantiated GTM with S_{old} to produce a candidate design for achieving F_{new}.

Case-Specific Device Models

Let us consider as an example the task of designing a Nitric Acid cooler (NAC_{new}) to reduce the temperature of some quantity of Nitric Acid from some initial temperature $T1$ to some final temperature $T2_{new}$. Let us also suppose that the case-retrieval task returns the design and model of a Nitric Acid cooler (NAC_{old}) which reduces the temperature of the same quantity of Nitric Acid from temperature $T1$ to temperature $T2_{old}$, where $T1 - T2_{new} >> T1 - T2_{old}$. Clearly, the desired function of cooling Nitric Acid from $T1$ to $T2_{new}$ is similar to but different from the delivered function of cooling Nitric Acid from $T1$ to $T2_{old}$. The difference between the two functional specifications, which we will denote as $F_{new} - F_{old}$, lies in the range by which the Nitric Acid is cooled.

The structure of NAC_{old} is shown in figure 2(a). It consists of a pump that pumps cold water into the device, a pipe through which hot Nitric Acid flows in the device, and a heat-exchange chamber which contains the cold water pumped into the device and includes the Nitric Acid pipe. The model for NAC_{old} specifies how the device works, i.e., how its structure delivers its function of cooling Nitric Acid from $T1_{old}$ to $T2_{old}$. The functioning of this device can be informally described as follows: Hot Nitric Acid flows through a pipe, a part of which is enclosed in a heat-exchange chamber. The chamber contains cold water that is pumped into the device by a water pump. Inside this chamber, heat is transferred from the hot Nitric Acid to the cold water. As a result of this transfer of heat, the temperature of out-flowing Nitric Acid is lower than the temperature of in-flowing Nitric Acid; the temperature of cold water increases correspondingly.

Kritik2 explicitly represents the functions, the structure and the *internal causal behaviors* of the device, where the internal causal behaviors specify how the device structure delivers its functions (Goel 1991). Its behavioral representation language generalizes the *functional representation scheme* (Sembugamoorthy & Chandrasekaran 1986) and grounds it in *component-substance ontology* of physical devices (Bylander & Chandrasekaran 1985). The internal causal behaviors in this language are represented as partially ordered sequences of states and state-

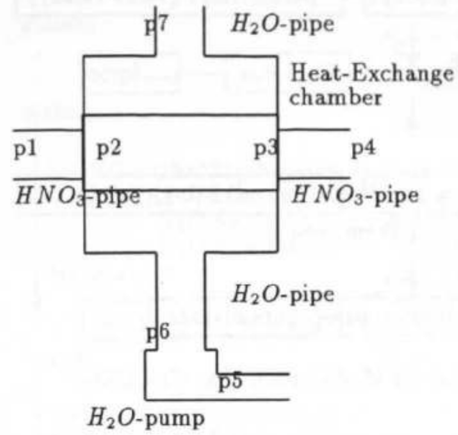

Figure 2(a): Structure of NAC_{old}

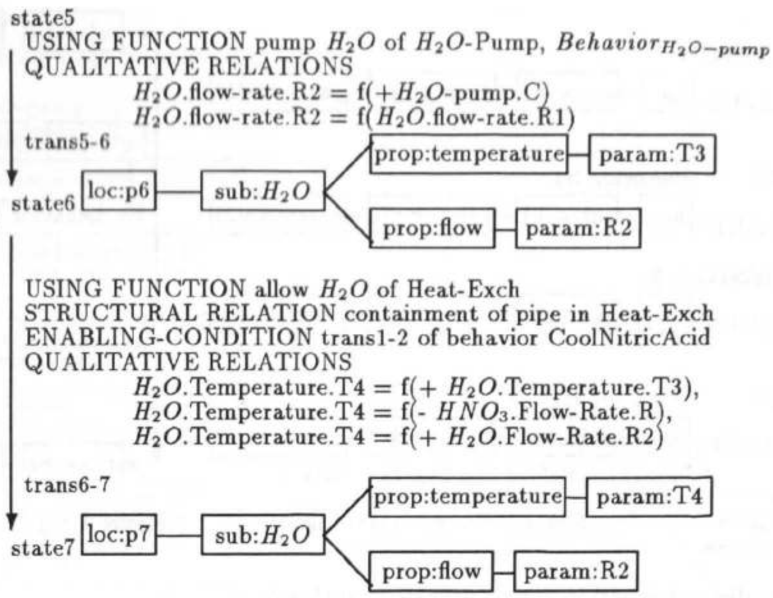

Figure 2(b): Behavior HeatWater of the old HNO_3 cooler,NAC_{old}

transitions. A state is a partial description of some substance of the device at some particular point in the device structure. A state transition describes how the parameters of a substance change as the substance flows from one point to another.

Figure 2(b) shows a fragment of an internal behavior of the Nitric Acid cooler called behavior **HeatWater**. This behavior describes the flow of the water through the device. Initially, at **state5**, water has a flow-rate R1, and temperature T3. In **state6**, the water flow-rate has increased to R2 due to the functionality of the water pump. After flowing through the heat-exchange chamber the water temperature is T4. The transition **trans6-7** is due to the multiple functions of the heat-exchange chamber, which allows the flow of the water and also allows the heat flow between hot Nitric Acid and cold water. This transition occurs simultaneously with **trans2-3** in the behavior **CoolNitricAcid** of the device, where behavior **CoolNitricAcid** specifies the state transitions of the Nitric Acid. Each transition is annotated with enabling conditions that need to be true in order for the transition to occur and with qualitative equations that relate the state changes. (Goel 1991) provides a more detailed description of case-specific device models.

Generic Teleological Mechanisms

Kritik2 posits a memory of GTMs such as cascading, feedback, and feedforward. The representation of a GTM encapsulates (i) knowledge about the difference between the functions of a known design and a desired design that the mechanism can help to reduce, and (ii) knowledge about modifications to the internal causal behaviors of the known design that are necessary in order to reduce this difference. A GTM thus associates a type of functional difference with a type of behavioral modification, with the former acting as an index to the latter.

Figure 3 illustrates Kritik2's representation of the cascading mechanism. Figure 3(a) specifies that the cascading mechanism is applicable when the known design (Design1) changes the value of some substance property from $val11$ to $val21$ by some known internal behavior B1, the desired design (Design2) changes the value of the same substance property from $val12$ to $val22$ by some Behavior B2, and $|val22 - val12|$ is many times $|val21 - val11|$. Figure 3(b) illustrates Kritik2's representation of the modifications necessary to reduce the functional difference. It specifies that Behavior B2 might be achieved by replicating Behavior B1 as many times as needed. Since, in general, $|val22 - val12|$ might not be a multiple of $|val21 - val11|$, Behavior B2 also includes the possibility of forming a new goal to reduce the functional difference left after replicating B1. Note that the behavioral model of the cascading mechanism is indexed by the functional difference it can reduce.

Given a specific type of functional difference between the desired design and the retrieved one, Kritik2 uses the functional difference to access the applicable GTM. For example, if the difference between the desired function and the delivered function is that the delivered function alters some substance property by some "small amount" and

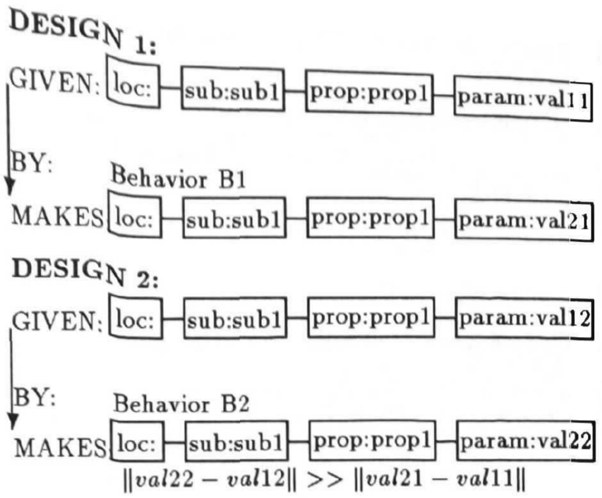

Figure 3(a): Functional Difference that Cascading reduces

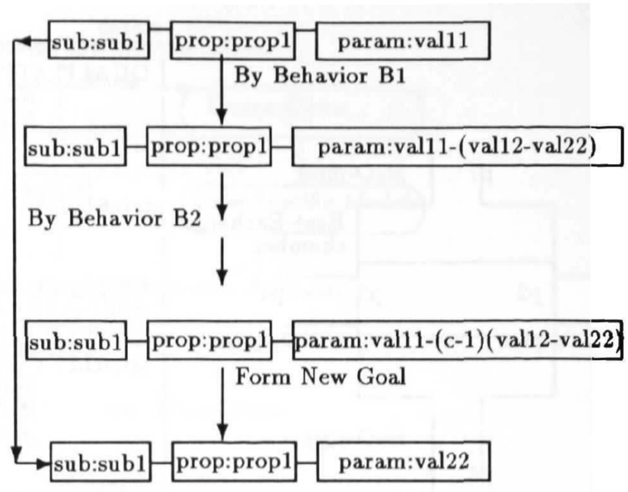

Figure 3(b): Behavior Modification that Cascading suggests

the desired function is the alteration of this property by a "large amount", then this functional difference can be used to access the GTM for cascading. Once accessed, the GTM can be applied to the internal causal behaviors of the known design.

Case Adaptation

Let us now consider how knowledge of the cascading mechanism helps in the task of adapting the structure of NAC_{old}, which reduces the temperature of the same quantity of Nitric Acid from temperature $T1$ to temperature $T2_{old}$, to design NAC_{new} which can cool the same quantity of Nitric Acid from $T1$ to $T2_{new}$, where $T1 - T2_{new} >> T1 - T2_{old}$.

Diagnosis : First, the diagnosis task identifies the set of structural elements that influence the substance properties that need to be changed, and the set of the specific behavioral state-transitions in which each element plays a role. In the above example the output of the diagnosis task is $S_{elements} = \{$ water-pump.C $\}$, where C is the capacity of the water pump. If $S_{elements}$ contains more than one element, then they can be ordered heuristically. (Stroulia, Shankar, Goel & Penberthy 1992) provide a more detailed description of the diagnosis step.

Repair : Next, the repair task instantiates the cascading mechanism in the context of the model of the known device. The repair plans in Kritik2 specify compiled sequences of operations that need to be performed for repairing a design, given a specific type of difference desired in its function. Kritik2 uses the difference $F_{new} - F_{old}$ as a probe into the memory of repair plans and retrieves the applicable plans. The candidate modifications $S_{elements}$ are used as a secondary index to discriminate among the applicable plans. Some repair plans contain pointers to GTMs. These plans also contain procedural knowledge of how to synthesize the behavior of a GTM with the model of the known device. If a repair plan of this type is selected, Kritik2 retrieves the appropriate GTM, instantiates it in the context of the model of old device, and synthesizes the behavior of the GTM with the model.

For our Nitric Acid cooler example, Kritik2 uses the type of functional difference between the desired and the retrieved designs as an index into the repair plan memory, and selects the repair plan called the ***structure-replication plan***. This plan contains a pointer to the cascading mechanism shown in Figure 3. It synthesizes the behavior of the cascading mechanism with the device model of NAC_{old} in two steps: (i) behavior revision and (ii) structure revision. First the behavior $B_{waterpump}$ is replicated in the internal behavior `HeatWater` shown in Figure 2(b) because the water pump was the structural element identified by the diagnosis task.

More specifically, since the water pump plays a role in **`trans5-6`** of behavior **`HeatWater`**, this transition is replicated to obtain the modified behavior `HeatWater` shown in figure 4(a). The changes in the values of state variables caused by this are propagated forward throughout the behavior. Since the changed values affect another behavior in the device model, namely, the behavior `CoolNitricAcid`, the values are propagated in this behavior as well.

Once the behavior revision is completed, the structure is revised. Since each state-transition explicitly specifies the structural elements which are responsible for the transition, the behavioral modifications are directly translated into structural modifications. The structure of the resulting design, with multiple water pumps, is shown in figure 4(b). Kritik2 now evaluates the candidate design by qualitatively simulating the case-specific device model. If the simulation reveals inconsistencies between the desired functions of the device and its output behaviors, redesign is needed.

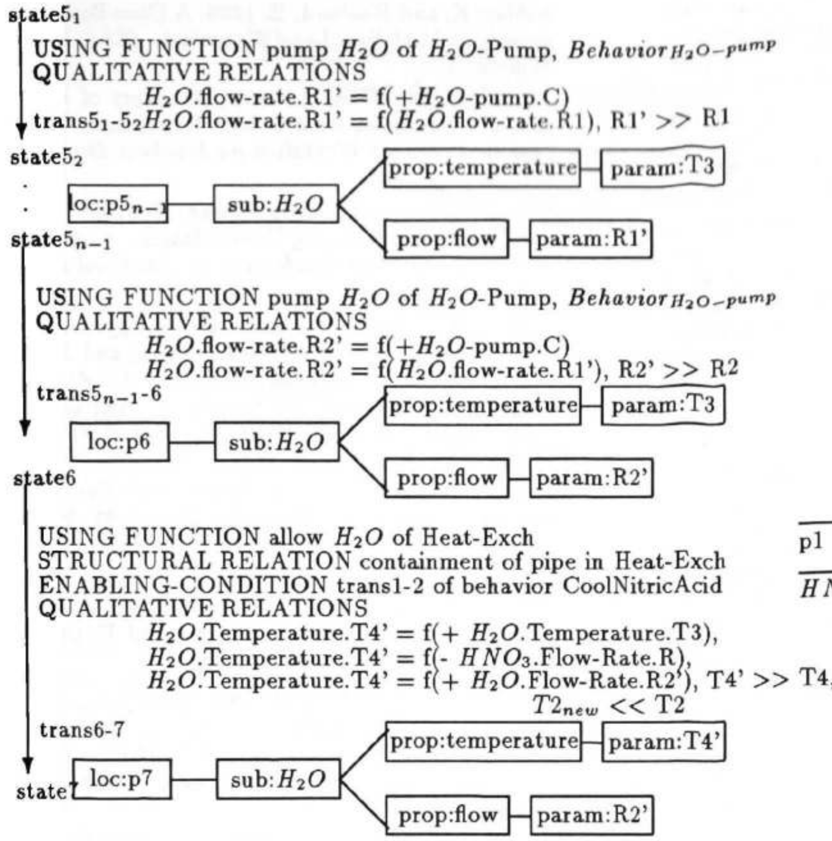

Figure 4(a): Revised Behavior HeatWater of the new HNO_3 cooler, NAC_{new}

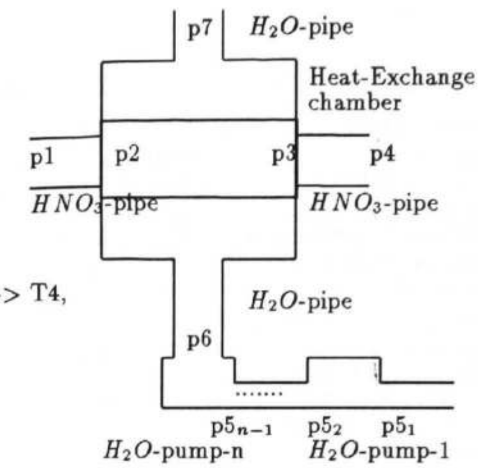

Figure 4(b): Revised Structure of the new HNO_3 cooler, NAC_{new}

Evaluation and Analysis

Kritik2 provides a computational testbed for conducting controlled experiments with GTMs and their use in case adaptation. The case memory in Kritik2 contains designs of four types of physical devices: simple heat exchangers of the type described above, electrical circuits such as the circuit in a household flashlight, electromagnetic devices such as the household buzzer, and complex angular momentum controllers such as those aboard the Hubble Space Telescope. This indicates that its component-substance ontology and behavioral representation language are not limited to any specific device domain. Kritik2 demonstrates the sufficiency of the scheme for representing, indexing, accessing, and using the cascading mechanism in two of these four domains: replication of pumps in heat exchangers and replication of batteries and resistors in electrical circuits. Again, this indicates that the method of using GTMs for case adaptation is not limited to any specific device domain. However, the present implementation of Kritik2 contains only the GTM for cascading. We are presently adding more mechanisms to our library of generic teleological mechanisms and evaluating them in more domains.

Ablation experiments with Kritik2, in which specific types of knowledge and methods of reasoning in the system are "lesioned" (Cohen & Howe 1988), indicate that its process model for use of GTMs in case adaptation is quite flexible. In general more than one repair plan in the plan memory may be applicable for a given case-adaptation task. For instance, the ***component-replacement*** plan is another repair plan applicable in our Nitric Acid cooler example. If selected, this plan probes the memory of components for a water-pump with higher capacity. If such a water-pump is available, the execution of this plan results in a simple substitution of one component (water-pump with a low capacity) by another (water-pump with a higher capacity). In that case, Kritik2 behaves like most previous case-based systems, in that it uses component replacement to tweak the retrieved design. If, however, a water-pump with a higher capacity is not available in the component memory, the component-replacement plan would fail. In this case, Kritik2 resorts to the use of the structure-replication plan which instantiates the cascading mechanism.

Additional ablation experiments with Kritik2 indicate a different kind of flexibility pertaining to the diagnosis task in the process model. Although the process model includes a diagnosis task, the diagnosis step actually is optional. The method of instantiating GTMs, however, results in poorer designs if the diagnosis task is not performed. In the Nitric Acid cooler example, for instance, we found that instantiating the cascading mechanism results in the replication of the water pump if the diagnosis task is performed, and in the replication of the entire heat exchanger if the diagnosis task is not performed. The former design is more parsimonious and hence better than the latter one. This leads us to conclude that while the quality of the solution appears to improve when diagnosis task is performed, instantiating GTMs appears to be a useful strategy for case adaptation whether or not the diagnosis task is performed.

Related and Further Research

Experience-based reasoning is a model of human decision making and problem solving (Riesbeck & Schank 1989). Previous work on experience-based reasoning has investigated the use of modification operators and rules for routine case adaptation in which the needed modifications involve changing the parameter of an element in the old solution or substituting one solution element by a similar one (Alterman 1988; Ashley & Rissland 1988; Hammond 1989; Kolodner & Simpson 1989). Exploration of more robust methods of case adaptation based on derivational traces (Carbonell 1986) and causal models (Goel 1991; Koton 1988; Sycara & Navinchandra 1989) also has been largely limited to relatively routine case adaptation.

Our research on generic teleological mechanisms builds on earlier research on model-based case adaptation. The Kritik2 system provides a model of how knowledge of GTMs might complement knowledge of case-specific device models and help designers to reason about patterned insertions of new components in old designs.

Darden has proposed that scientific theories can be viewed as devices and theory revision can be viewed as a design-adaptation task (Darden 1990). In recent personal communication (Darden 1991), she has further conjectured that Kritik2's use of GTMs for design adaptation might provide a basis for modeling the formation of early theories of heredity and genetics. If this is correct, it would indicate that the use of GTMs is a very general domain-independent method of case adaptation. Our current work on GTMs involves modeling how designers learn GTMs from specific design cases and use this knowledge in analogical reasoning across different domains (Bhatta & Goel 1992).

Acknowledgements We wish to thank Sambasiva Bhatta for many discussions on the subject and comments on earlier drafts of this paper.

References

Alterman, R. 1988. Adaptive Planning. *Cognitive Science*, 12:393-422.

Ashley, K. and Rissland, E. 1988. A Case-Based Approach to Modeling Legal Expertise. *IEEE Expert*, Summer 1988.

Bhatta, S. and Goel, A. 1992. Discovery of Principles and Processes from Design Experience. To appear in *Procs. of Workshop on Machine Discovery*, ML-92.

Bylander, T. and Chandrasekaran, B. 1985. Understanding Behavior Using Consolidation. *Proc. Ninth International Joint Conference on Artificial Intelligence*, 450-454.

Carbonell, J. 1986. Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. *Machine Learning: An Artificial Intelligence Approach, Volume II*, R. Michalski, J. Carbonell and T. Mitchell (editors). San Mateo, CA: Morgan Kauffman.

Cohen, P. and Howe, A. 1988. How Evaluation Guides Research. *AI Magazine*, 9(4):35-43, Winter 1988.

Darden, L. 1990. Finding and Fixing Faults in Scientific Theories. *Computational Models of Discovery and Theory Formation*, J. Shrager and P. Langley (editors). Hillsdale, NJ: Erlbaum.

Darden, L. 1991. Personal Communication.

Goel, A. A Model-Based Approach to Case Adaptation. *Proc. Thirteenth Annual Conference of the Cognitive Science Society*, Chicago, August 7-10, 1991, pp. 143-148.

Hammond, K. 1989. *Case-based Planning: Viewing Planning as a Memory Task*, Boston, MA: Academic Press.

Kolodner, J.L. and Simpson, R. 1989. The MEDIATOR: Analysis of an Early Case-Based Reasoner. *Cognitive Science*, 13:507-550.

Koton, P. 1988. Combining Causal and Case-Based Reasoning. *Proc. Tenth Annual Conference of the Cognitive Science Society*.

Riesbeck, C. and Schank, R. 1989. *Inside Case-based Reasoning*. Hillsdale, NJ: Erlbaum.

Sembugamoorthy, V. and Chandrasekaran, B. 1986. Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems. In *Experience, Memory and Reasoning*, J. Kolodner and C. Riesbeck: (editors), Hillsdale, NJ: Lawrence Erlbaum, pp. 47-73.

Stroulia, E., Shankar, M., Goel, A. and Penberthy, L. 1992. A Model-Based Approach to Blame Assignment in Design. To appear in *Proc. Second International Conference on AI in Design*, Pittsburg, June 1992.

Sycara, K. and Navinchandra D. 1989. A Process Model of Case-Based Design. *Proc. Eleventh Cognitive Science Society Conference*.