

Behavioral Synthesis in CADET, a Case-Based Design Tool

D. Navinchandra, Katia P. Sycara, S. Narasimhan

School of Computer Science
Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

Good mechanical designs are often composed of highly integrated and interacting components. Prior designs represent good solutions to the tightly coupled nature of mechanical devices; case pieces already embody good mechanical design principles and advantageous incidental component behaviors. This makes case based reasoning a more appropriate design methodology than a priori decomposition and recombination techniques. In addition, combinations of independent standard components typically result in suboptimal designs. It is, however, difficult to recognize and retrieve case pieces that realize subfunctions of a device, given its specifications. We present an approach to transforming the given specifications in a behavior-preserving manner, so as to enable retrieval, and subsequent synthesis of relevant component cases and pieces of cases.

AI Topic: AI Applications to Design, Case-Based Reasoning. **Domain Area:** Engineering Design of Hydro-mechanical Devices. **Language/Tools:** CommonLisp, Knowledge-Craft and the X window system. **Status:** Research Prototype. **Effort:** 1.5 person years. **Impact:** Provides, for the first time, support in synthesizing novel designs from a memory of prior designs. The current case-base has 20 Hydro-mechanical devices. **Support:** DARPA and AFOSR under contract F49620-90-C-0003.

Introduction

The process of satisfying design specifications can be viewed in two ways [1]. First, via an *a posteriori* process: by testing whether a candidate design satisfies its specifications. This process usually involves simulation, evaluation, and constraint checking. Second, via an *a priori* process: by eliciting or constraining the generation of possible design structures to be consistent with the set of specifications. The second process amounts to reasoning from goals and constraints to (parts of) designs. One way to operationalize this process in the domain of mechanical systems is to use transformations that convert

the given specifications into alternate forms that facilitate the identification of relevant components and the subsequent realization of the design. We discuss desirable characteristics of these transformations and present an approach to transform functional specifications of a device into behaviorally equivalent forms.

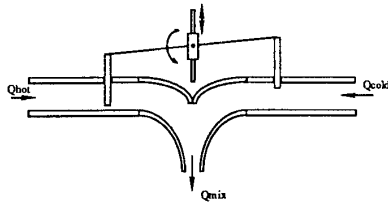
Behavior preserving transformations require formal representations of the behavioral specifications of mechanical systems as well as formal representations of behavioral characteristics of mechanical components. The representation that we use is the language of qualitative physics [5]. Qualitative physics has predominantly been used in analyzing the behavior of existing systems. It provides a general representational scheme for classes of mechanical behavior. The transformational approach we propose decomposes given behavior specifications into "sub-behaviors" without altering the overall desired behavior. The decompositions do not impose any a priori structure or topology on the physical realization of the design. The decompositions are collections of sub-behaviors with information on how the sub-behaviors must interact to produce the overall device behavior. In effect, we do not require an a priori decomposition of design specifications.

Due to the tightly coupled and interacting nature of mechanical designs, reasoning from past designs to synthesize a new design is a suitable methodology as opposed to direct "decompose and recombine" strategies. This is because design cases can reflect good design principles [6, 3], such as function sharing [8] and may incorporate decisions that take advantage of, or compensate for incidental components interactions. Our investigation of the proposed behavior preserving transformational strategies is conducted within a framework of a case based reasoning methodology for mechanical design and has been implemented in the CADET (Case Design Tool) system. CADET operates in the domain of hydro-mechanical devices, such as faucets, flush tanks, valves, and pumps.

The cases have associated descriptions both in terms of

behavioral and structural characteristics. If parts of the behavioral specifications of the desired design correspond to the behavioral indices of the components, then the components (and hence their structural description) can be directly accessed. However, as there is no one-to-one correspondence between the desired behavior of a device and the individual component behaviors, it is often not possible to find relevant cases by using the given overall behavioral specification as an index into case memory. This gives rise to the need for behavior-preserving transformation techniques to transform an abstract description of the desired behavior of the device into a description that gives rise to indices with which relevant components can be retrieved.

For example, consider the design of a device which controls the flow of water into a flush tank. The behavior can be specified as follows: *as the depth of water (D) in the tank increases, the rate of flow of water into the tank (Q) should decrease.* This specification may be used as a set of indices to find relevant cases in memory. If there are no cases which directly match the specifications, then it would be useful to consider using parts of cases if possible. In this instance, a relevant case could be a hot-cold water faucet shown in the figure below. The faucet is specified as a device which allows for the independent control of the temperature and flow rate of water by appropriately mixing hot and cold water streams. The relevance of the faucet case to the design of the flush tank would not have been possible to recognize given the initial specifications of the two devices. However, by extracting portions of the faucet, such as the see-saw part, it is possible to design the flush tank device as shown in Figure 1.

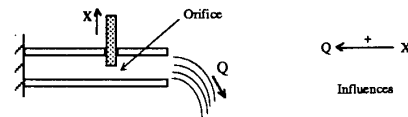


CADET transforms behavioral descriptions by opportunistically deciding to either hypothesize new behaviors or to rely on known laws. This is in contrast to other approaches that assume *a priori* knowledge of the domain laws and models that will be part of the solution [11]. If CADET cannot complete a design because it is not given the relevant physical laws, it hypothesizes behaviors and looks for cases which embody the relevant laws. Our approach has the following advantages: (1) the system at each point in the search is aware of what behavior it is trying to achieve, (2) because cases embody design optimizations, the accessed components correspond to al-

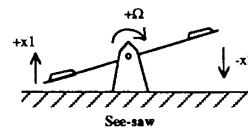
ready optimized physical structures, (3) solutions may involve use of principles outside the given domain, that have been successful in a prior design, (4) the problem solver does not have to re-solve problem from scratch.

Representation

In CADET, device behavior is represented as a collection of influences among the various inputs and outputs [4, 9]. An assumption we make is that the process already exists. We do not consider devices which create processes. An influence is a qualitative differential (partial or total) *relation* between two variables one of which is a dependent variable and the other an independent variable. The notion of influences is based on the notion of confluences [2]. Influences are organized as graphs. In general, an influence graph is a graph whose nodes represent the variables of concern and whose arcs are qualitative values, depicting the influence of one variable on another. For example, a household water tap has two inputs: a water source and a signal to regulate the rate of flow of water. In the two and a half dimensional representation [10] that we have adopted, the tap is represented as a pipe with a gate as shown below. The flow rate is given by Q and the position of the gate is given by X . The position of the gate controls the flow rate. This behavior is represented as an influence $Q \leftarrow^+ X$, which is read as follows: "The flow rate (Q) increases (+) monotonically with an increase in the signal (X)".



Sets of qualitative influences can be combined to capture the behavior of more complex devices. The see-saw shown below has three major behavioral parameters: Ω , the angular position of the see-saw and the positions of the two ends of the see-saw ($X1$ and $X2$). The main influences are $X1 \leftarrow^+ \Omega$, $X2 \leftarrow^- \Omega$ and $X2 \leftarrow^- X1$. These influences form a directed graph.



In CADET, influence graphs are used to represent the behavior of devices. When this representation is incorporated in a device case base, it becomes possible to retrieve cases which match given behavior specifications. If retrieval using the design specification fails to retrieve relevant cases, the system should be able to recognize how a combination of component behaviors could produce the re-

quired effect.

Transformation

Returning to the flush tank, the behavior can be specified as follows: as the depth of water (D) in the tank increases, the rate of flow of water into the tank (Q) should decrease. The influence is given by $Q \leftarrow^- D$. A possible solution to the problem is shown in Figure 1. Three cases have been used: the see-saw, the tap, and a float. The solution consists of the following influences: $X1 \leftarrow^+ D$, $X2 \leftarrow^- X1$ and $Q \leftarrow^- X2$ which correspond to the float, see-saw, and tap respectively. The combined effect of these three influences is equivalent to the original specification. To find these relevant cases, the system should be able to deduce that the original specification can be decomposed into three influences that can match the cases.

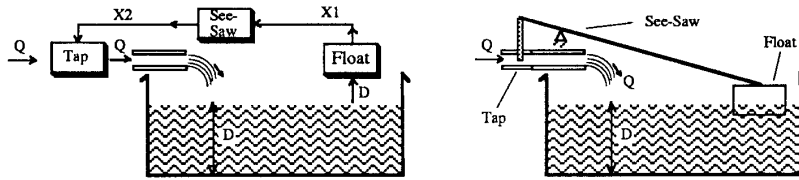


Figure 1: A flush tank

Two Rules for Propagating Influences

1. *The function of function (chain) rule.* Let $u = f(x, y)$, where x and y are independent quantitative variables with respect to u . Let $z = g(u, v)$ be another quantitative function. Then, $\left[\frac{\partial z}{\partial x} \right] = \left[\frac{\partial z}{\partial u} \right] \cdot \left[\frac{\partial u}{\partial x} \right]$

The square brackets ([]) indicate a qualitative operator. The operator returns +ve, -ve or 0 when the numerical value of the expression within the brackets is more than, less than or equal to zero respectively.

2. *The total influence rule.* Let $u = f(x, y)$ be a continuous quantitative function. Then the total influence $[\Delta u] = \left[\frac{\partial u}{\partial x} \right] \times [\Delta x] + \left[\frac{\partial u}{\partial y} \right] \times [\Delta y]$. This gives the net qualitative increment in the dependent variable as the independent variables are changed.

These two rules can be applied to influence graphs to derive new influences. For example, consider a kitchen sink with two taps pouring water into it (Figure 2). The flow rates of the two streams of water are $Q1$ and $Q2$ and the depth of water in the sink (at equilibrium) is D . The influences on D are: $D \leftarrow^{Op1} Q1$, $D \leftarrow^{Op2} Q2$, where $Op1$ and $Op2$ are both positive influences. The rate of flow out of the sink ($Q3$) is influenced by the depth of water D : $Q3 \leftarrow^{Op3} D$ where, $Op3$ is positive.

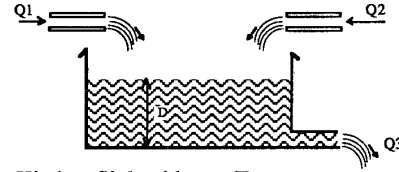
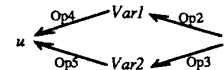


Figure 2: Kitchen Sink with two Taps

From the function of function rule we can conclude that the input flow $Q1$ will influence the total outflow $Q3$ as $Q3 \leftarrow^{Op1 \times Op3} Q1$. In other words, an increase in $Q1$ will cause the rate of flow through the drain $Q3$ to increase. The sign of the influence is the qualitative multiplication of $Op1$ and $Op3$. These two rules, when applied in reverse, become design rules:

1. *Inverse application of the function of function rule:* If the influence $z \leftarrow^{Op1} x$ is given as a goal, then the goal may be achieved by introducing an unknown intermediate variable $Var1$ to give the influences: $z \leftarrow^{Op3} Var1 \leftarrow^{Op2} x$. Where, $Op3 \times Op2 = Op1$. This follows from the original Function of Function rule. The two new influences serve as an elaboration of the original influence to which they are behaviorally equivalent.

2. *Inverse application of the total differential rule:* If the influence $u \leftarrow^{Op1} v$ is given as a goal, then the goal may be achieved by introducing two unknown intermediate variables $Var1$ and $Var2$ to give the influences:



Where, $Op4 \times Op2 + Op5 \times Op3 = Op1$.

The two operators are systematically applied to a given influence to yield complex transformations. These transformations can then be used to find matching cases. Hence, even though it might not be possible to find a case which matches the original specification, it may be possible to find several cases which match a transformation of the original specification. For example, the original flush tank specification can be transformed as follows: two serial transformations of the influence, ($Q \leftarrow^- D$) yields

the following three sets of influences:

1. $Q \leftarrow^{-} Var2 \leftarrow^{+} Var1 \leftarrow^{+} D$
2. $Q \leftarrow^{+} Var2 \leftarrow^{-} Var1 \leftarrow^{+} D$
3. $Q \leftarrow^{+} Var2 \leftarrow^{+} Var1 \leftarrow^{-} D$

The second set of influences (#2. above) will match the tap, the see-saw and the float cases respectively.

Using Domain Laws and Principles.

Domain laws are used as the basis for elaborating influences. This approach is preferred to just hypothesizing intermediate variables (using the two elaboration rules) and then unifying the unknown variables. The first step is to determine the influences implied by the domain laws. This is done by symbolically differentiating the equations which represent the laws. Next, the two rules are applied as follows:

Rule 1: If the influence $z \xleftarrow{Op1} x$ is given as a goal, and if it is known from a domain law that some quantity A is known to influence z : ($z \xleftarrow{Op3} A$), then the goal may be achieved by having x influence A and hence indirectly influences z : ($z \xleftarrow{Op3} A \xleftarrow{Op2} x$), where $Op3 \times Op2 = Op1$.

Rule 2: If the influence $u \xleftarrow{Op1} v$ is given as a goal, and if it is known from a domain law that two quantities A and B influence u such that: $u \xleftarrow{Op4} A$ and $u \xleftarrow{Op5} B$, then the goal could be achieved by making v influence A or B , or both.

Reconsider the kitchen sink example. Let's say we want to increase the total outflow $Q3$ with respect to some external signal Sig . The required influence is $Q3 \xleftarrow{+} Sig$. This influence can be transformed by finding what factors influence $Q3$. From the *Law of Conservation of Mass* we know that: $Q3 = Q1 + Q2$ at equilibrium. Using symbolic calculus the following influences are derived from the law: $Q3 \xleftarrow{+} Q1$, $Q3 \xleftarrow{+} Q2$. Influences between $Q1$ and $Q2$ are not used as they are specified to be independent variables. It follows that the variable Sig can be made to influence $Q1$ or $Q2$ in order to influence $Q3$ indirectly. There are three possible influences now: (a) the original influence $Q3 \xleftarrow{+} Sig$, (b) the first indirect influence: $Q3 \xleftarrow{+} Q1$, $Q1 \xleftarrow{+} Sig$ and (c) the second indirect influence: $Q3 \xleftarrow{+} Q2$, $Q2 \xleftarrow{+} Sig$. Combinations of the above influences yields possible transformations.

A Conceptual Design Example

In this section, we discuss an example that illustrates the behavior of the CADET system. The example is about the design of a hot/cold water faucet. The function of the faucet is described as:

A device which *mixes* hot water at temperature T_h and cold water at temperature T_c with flow rates Q_h and Q_c

respectively and allows the control of the mixed water temperature T_m by a mechanical signal S_t and its flow rate Q_m and by a mechanical signal S_f . In addition, the two controls should be independent: S_t should not influence Q_m and S_f should not influence T_m .

This specification is input to CADET in the form of goals, constraints and a qualitative description of the governing physical laws and principles.

Goals. The goal of the system is to select a set of components which take a certain input and produce a certain output. These goals can be represented using influences

$$\text{as: } \left[\frac{\partial T_m}{\partial S_t} \right] = [+], \quad \left[\frac{\partial Q_m}{\partial S_f} \right] = [+]$$

Constraints. For the faucet, the behavior specification says that the signal S_t should not influence Q_m and the signal S_f should not affect T_m . For CADET the above constraints are input as:

$$\left[\frac{\partial T_m}{\partial S_f} \right] \neq [+], \quad \left[\frac{\partial T_m}{\partial S_t} \right] \neq [-], \quad \left[\frac{\partial Q_m}{\partial S_t} \right] \neq [+], \quad \left[\frac{\partial Q_m}{\partial S_f} \right] \neq [-]$$

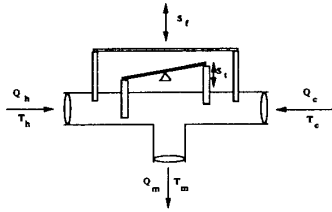
Process description. For the faucet the process controlled by the device is the mixture of fluid flow. The behavior of the mixture can be described in terms of the following two equations: (1) $Q_m T_m = Q_h T_h + Q_c T_c$ (Conservation of Energy), and (2) $Q_m = Q_c + Q_h$ (Conservation of Mass), where, $T_h > T_c$.

The Design Process

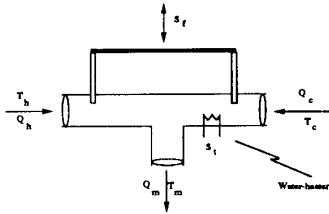
CADET first tries to find cases which directly match the given goal. If it fails, it starts elaboration by alternatively applying the two rules. Before it starts hypothesizing variables, it tries to elaborate using the known domain laws. Here are some of the alternatives generated (the influences are written in as lists):

1. ($T_c S_t +$) ($Q_c S_f +$) ($Q_c S_t -$) ($Q_c S_f +$)
2. ($Q_m S_t -$) ($T_m S_t +$) ($Q_m S_f +$) ($T_m S_f +$)
($Q_m S_t +$) ($T_m S_t +$) ($Q_m S_f +$) ($T_m S_f -$)
3. ($T_c S_t +$) ($Q_c S_f +$) ($T_c S_t +$) ($Q_h S_f +$)
4. ($T_m S_t +$) ($Q_m S_f +$)

The first solution is rejected as it violates the given constraints. In the second solution we find that S_t influences T_m positively, but influences Q_m both negatively and positively. CADET assumes that S_t will not influence Q_m if the quantitative increase and decrease are equal. Similarly, we find that S_f will influence Q_m positively, but can be made not to influence T_m . The potential design, based on the cases retrieved for this elaborated index is shown below.



The third solution is shown below. Note that this device cannot deliver pure cold water. Though the original behavior description is satisfied, the design does not work properly at the limits. This is due to CADET's inability to handle non-continuous behavior.



The fourth solution is the the initial specification itself. The solution involves just a tap and a water heater/cooler placed at the outlet of the T-Pipe.

Concluding Remarks

We have presented an approach to the conceptual design of hydro-mechanical systems using a case base of previous designs. The process consists of applying behavior-preserving transformations, based on a qualitative calculus, to an abstract description of the desired behavior of the device until a description is found that closely corresponds to some collection of relevant cases. The major benefits of the approach are: (a) it allows for retrieval of relevant cases without imposing a predetermined decomposition of the design, (b) it is a generative approach that utilizes knowledge of domain laws, design principles, such as simplicity, and behavioral constraints to reason from design goals to possible solution structures, (c) it can identify "missing" cases, necessary for completion of the design, and (d) the resulting transformations are guaranteed to be behaviorally equivalent to the original specification.

Future works will focus on extending CADET's capability to reason about feedback and intermittent devices [7]. In feedback systems preserving behavior during index transformation is not easy.

References

[1] Amarel, S. . Presentation at Knowledge Based Systems Workshop, Los Angeles, CA.

- [2] DeKleer, J., J. Brown . "A Qualitative Physics Based on Confluences". *Artificial Intelligence* 24 (1984).
- [3] Gero, J., M. L. Maher, W. Zhang. Prototypes in Design. In *Proceedings of the 3rd AI in Engineering Conference*, Gero, J., Ed., Computational Mechanics, U.K., 1988.
- [4] Goel, A., B. Chandrasekaran. Functional Representation of Designs and Redesign Problem Solving. *Proceedings of the Eleventh (IJCAI) International Joint Conference on Artificial Intelligence*, 1989, pp. 1399-1394.
- [5] Kuipers, B. "Commonsense Reasoning about Causality". *Artificial Intelligence* 24 (1984).
- [6] Maher, M.L., F. Zhao. Using Experience to Plan the Synthesis of New Designs. *Engineering Design Research Center, CMU*, 1986.
- [7] Pu, Pearl. *Qualitative simulation of ordinary and intermittent mechanisms*. Ph.D. Th., Dept. of Computer Science, Univ. Of Pennsylvania, 89.
- [8] Suh, N.P., A.C. Bell, D.C. Gossard. "On an Axiomatic Approach to Manufacturing and Manufacturing Systems". *Journal of Engineering for Industry* (May 1978).
- [9] Sycara, K. and Navinchandra, D. Representing and Indexing Design Cases. *Proceedings of the Second International Conference on Industrial and Engineering Applications of AI and Expert Systems*, Tullahoma, TN., 1989.
- [10] Ulrich, K. T. and Seering, W. P. Function Sharing in Mechanical Design. *7th National Conference on Artificial Intelligence*, Minneapolis, MN, August 21-26, 1988.
- [11] Williams, B. Interaction-Based Invention: Designing Novel Devices from First Principles. *Proceedings of AAAI-90*, 1990, pp. 349-356.