Qualitative Reasoning about Function:
A progress report

Kenneth D. Forbus
Qualitative Reasoning Group
The Institute for the Learning Sciences
Northwestern University

Abstract: One important kind of reasoning in engineering problem solving is reasoning about function. This paper outlines research underway in our group to develop an account of reasoning about function, based on ideas from qualitative physics and motivated by teaching and design tasks. Our approach is based on three ideas: that the semantics of function representations should be based on behaviors, that qualitative and qualitative models of function play a central role in engineering reasoning, and that understanding the relationship between function and modeling assumptions is critical for many engineering reasoning tasks.

1. Introduction

Engineers design systems to achieve particular purposes. Verifying that a design will in fact behave in such a manner as to achieve its purpose is an example of *reasoning about function*. Another example is explaining the role a subsystem plays in a larger structure. For instance, the condenser in a steam propulsion plant serves both to reject waste heat to the environment and to recycle condensate to serve as feedwater, thus reducing the load on the ship's distillation plant. A third example is finding work-arounds, ways to keep a system operating under degraded conditions. Driving down a mountain road with a leaky brake cylinder, for example, may require pumping the brake to slow down, and periodically stopping to replenish the fluid in the brake system.

These examples suggest that reasoning about function is important for many engineering tasks. This has been recognized by many AI researchers, ranging from the work of Chandrasekaran and his students (cf. [1]) in using functional representations in diagnosis to de Kleer's early work on understanding the teleology of electronic circuits using qualitative reasoning [2]. Recently there has been renewed interest in reasoning about function, motivated both by progress in representing and reasoning about behaviors and by AI workers tackling more challenging problems. This paper describes work in progress by our group on representing and reasoning about function. Section 2 outlines our approach. Sections 3 and 4 describe two tasks which provide the context for our work. Section 5 summarizes.

2. Our Approach

Our goal is to extend the state of the art in qualitative physics to include representations of function, and reasoning techniques which use these representations to capture more of the kind of reasoning about physical artifacts performed by scientists, engineers, and just plain folks. Since this research is occurring in the context of a long-term effort to develop a knowledge base for engineering thermodynamics, we restrict ourselves to systems of that kind, along with the pneumatic and/or hydraulic control systems used with them. We measure our progress with reference to two tasks:

- *Education and Training:* We want an account of functional representations and reasoning which can be used in the construction of teaching software, including intelligent tutoring systems and learning environments.
- *Design:* Our account should provide leverage in building computer programs which can help designers in new ways, such as critiquing designs and ensuring that specifications are correctly met.

Our particular approach is motivated by the work of Franke [3], who showed that at least some funtional predicates could be given precise definitions in terms of behaviors. For example, a part `P` of a system `S` can be said to `PREVENT` some behavior `B` if (a) the set of possible behaviors for `S` does not contain `B` and (b) the set of possible behaviors of a new system `S'` which differs from `S` in that `P` is removed does include `B` in its set of possible behaviors. A relief valve in a boiler, for instance, prevents the boiler from reaching a pressure so high that the walls of the boiler would rupture. Recently Vescovi, Iwasaki, Fikes, and Chandrasekaran [4] extended this approach to provide causal accounts that are rich enough to assign credit for particular aspects of behaviors to individual model fragments. Their language, CFRL, provides a vocabulary for defining functional concepts in terms of properties of behaviors. Since CFRL is based on compositional modeling [5], which is a generalization of ideas from Qualitative Process theory [6], we plan to use CFRL as a starting point, extending it with representations from QP theory as needed.

Here is a summary of some of the extensions and new directions we are investigating:

- We believe our logic of occurrence [7] provides a more natural vocabulary for behaviors than those used by existing function representation systems. The reason is that for complex systems with many modes of operations, envisionments provide a better conceptual foundation than attainable envisionments (as used in Franke's work, since it exploits Kuiper's QSIM formalism [8]) or forcing behaviors to be linear sequences of states (as CFRL does). This does not mean that computing total envisionments is necessary to reason about function, of course. The important properties which must be captured by the formalism are that (1) branching can occur and (2) the starting state of a system being analyzed is often incompletely specified, which requires a means of characterizing the kinds of behaviors that are possible starting points.

- We believe that many functional terms require ontological commitments. For example, `TRANSFER` and `CONVERT` each define patterns of change over some kind of stuff. `TRANSFER` can be partially defined in terms of QP theory as a pattern of direct influences which conserve the total amount of some class of quantity. For instance, the following influences

```
I+(amount-of-in(oil,liquid,sump),pump-flow-rate)

I-(amount-of-in(oil.liquid,cooler), pump-flow-rate)
```

suggests a transfer of oil from the cooler to the sump, assuming that the quantity amount-of-in is an extensive parameter which defines a stuff. If the quantities involved are different, e.g., `internal-energy` and `amount-of-in` in a situation where combustion is occurring, `CONVERT` can be said to occur.

- An important aspect of reasoning about function is identifying and validating the assumptions underlying the models used in reasoning. Consider for example the problem of recognizing the function of an artifact based on an analysis of its likely behaviors. Given that a particular physical phenomena can typically be modeled at several levels of detail, the right choice of model can be important in figuring out how the artifact works. Figuring out the purpose of an automatic combustion control system, for instance, is impossible if the boiler has been modeled as an infinite source of steam.

The next sections describe the specific projects we are pursuing to investigate these ideas.

3. Reasoning about Control Systems

Control systems are, in some sense, function incarnate. Control theory provides a vocabulary of ideas, such as measurement, comparison, and correction, that constitute an essential part of every engineer's vocabulary. Formalizing this vocabulary is therefore a natural place to start in investigating reasoning about function. Furthermore, a formal vocabulary of control concepts could have many useful applications. Consider for example the training problems at the U.S. Navy's Surface Warfare Officer's School, which trains officers in the operation and maintenance of shipboard propulsion systems. As it happens, Automatic Boiler control systems are among the most complex systems in a propulsion plant. Students find them very difficult to understand. Instructors find them very difficult to teach. Yet they are critical for efficient plant operation. To build intelligent tutoring systems and learning environments, we must first develop good representations for control concepts.

One problem with the way training currently occurs is a focus on physical phenomena and representations. Consider for example the Automatic Combustion Control (ACC) system, which is part of the Automatic Boiler Control system. The purpose of the ACC is to maintain an appropriate steam pressure and fuel-air ratio in the face of changing demands for steam. It does this by measuring steam pressure, steam flow, and air flow, and adjusting the flow of fuel and air accordingly. Some of the complications in the design of an ACC include the following:

- The fuel-air ratio must be kept within a narrow range -- too rich, and visible black exhaust will be generated, which is undesirable if one wishes not to be seen. Too lean, and white smoke will be produced, which signals loss of efficiency.

- The air flow is controlled by Forced Draft Blowers (essentially, huge fans) whose inertia is significant. Consequently, there is a maximum rate at which their speed can be changed.

Physically, Automatic Combustion Controls can be complicated devices: A typical schematic in a Navy manual consists of three sensors, nine pneumatic computing elements which perform comparisons, selections, and scaling, and two controlled elements. In some pneumatic implementation technologies, the computing elements are the equivalent of operational amplifiers, where the same kind of element is used to implement multiple functions according to the equipment it is connected to. The relationships between the physical structure and the function can be quite complex: Sometimes a single aspect of function is distributed among several components, while in other cases a single component implements several distinct aspects of function. For example, the device which measures the pressure of steam in a boiler, the Master Sender, provides the measurement, comparison, proportional band, and reset action for that boiler, producing as output a signal which is then processed by several components to produce the appropriate control actions.

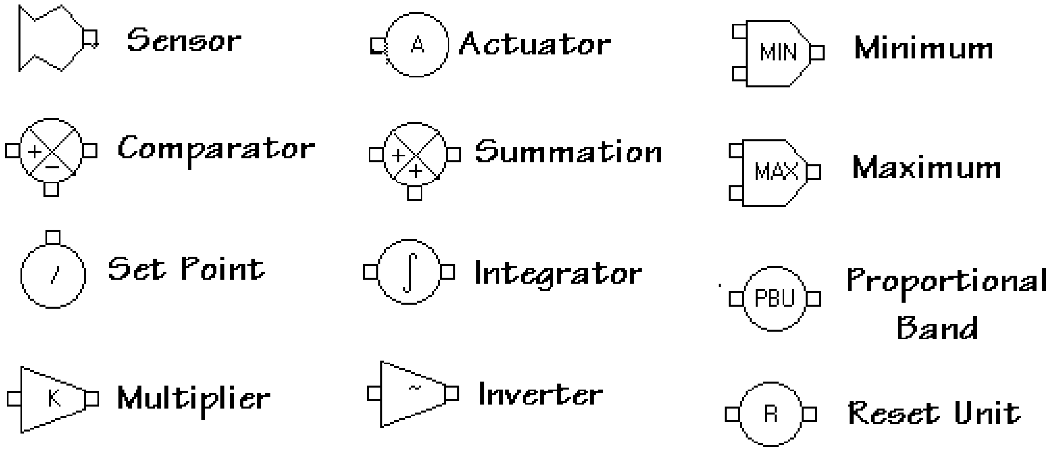


Figure 1: Graphical notation for feedback vocabulary used by the Feedback MiniLab

Even though students must ultimately understand how the components of this system implement its functionality, we suspect that making sure students are first well-versed in the ideas of feedback control at a functional level will accelerate their learning. Consequently, we have been developing a functional vocabulary for feedback systems, and a program which allows students to investigate feedback ideas with a functional representation, without regard to an underlying implementation technology.[1] Figure 1 shows the graphical notation for the vocabulary. The current program allows students to use a graphical editor to create and experiment with controllers via numerical simulation. Figure 2 illustrates one such controller, for a single input, single output control problem (i.e., keeping the level of fluid in a tank constant with variable flow in by manipulating the outlet flow).

Notice that the representation used, although functional, is numerical rather than qualitative. Automatically constructing simulated controllers is much simpler than building general physical simulations, because the flow of information inside the controller lends itself to a straightforward object-oriented simulation. Currently we are working towards extending this system in two ways:

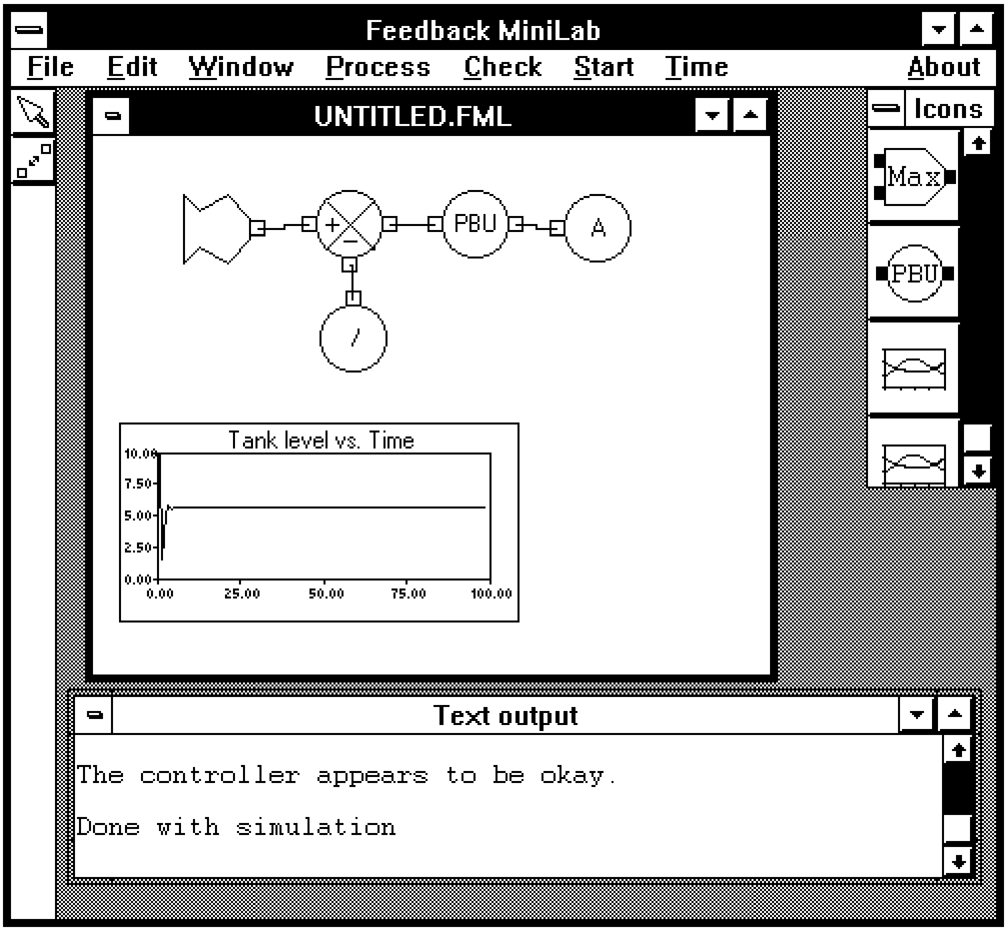
1. We are developing a qualitative domain theory to complement the numerical models that currently implement the functional vocabulary. By supporting qualitative reasoning about control systems, we should be able to build programs that provide critiques of a student's design.

2. We are extending the Feedback MiniLab to use self-explanatory simulators [10,11] as system simulations. Using self-explanatory simulators should both improve the quality of explanations and support more interesting student critiques. For example, if the qualitative analysis predicts that the student's controller won't correct properly

---

[1] The initial version of this system was written as part of the STEAMER project [9].

for changes in a particular parameter, the self-explanatory simulator could be used to search for a set of disturbances that would bring that situation about and show it to the student.



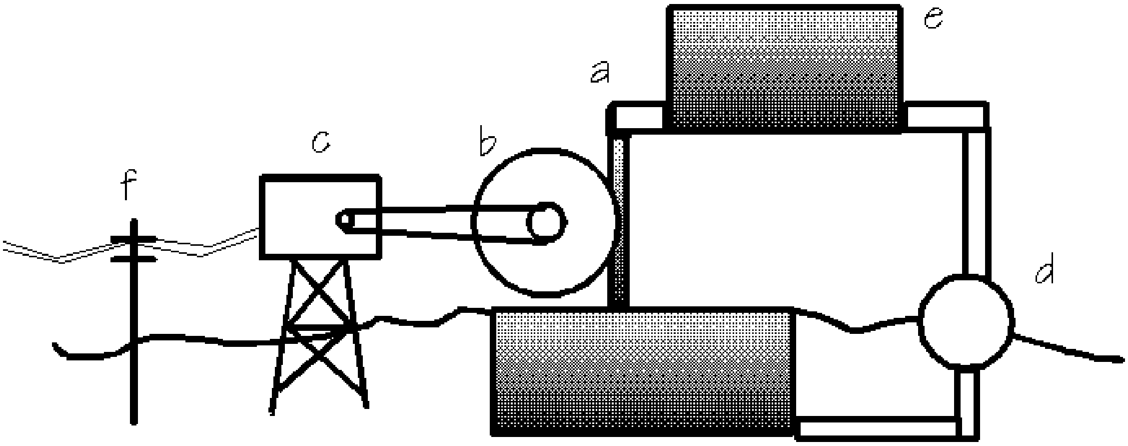
```
The controller appears to be okay.

Done with simulation
```

Figure 2: A controller built using the Feedback MiniLab

4. Skeptical Analysis of Designs

An important subproblem in design is verifying that an artifact can actually achieve its intended function. In some ways this is simpler than the problem of recognizing function, because the presumed function is part of the input. In other ways it is more complex, since the degree of accuracy desired can range from purely qualitative (e.g., using an incremental envisioner to demonstrate that the desired behavior is possible) to detailed numerical values (e.g., deriving via analysis or numerical simulation a set of ranges for parameter values that constrain the artifact to exhibit the desired behaviors). Another source of complexity is that the design may only have been partially analyzed, or, in the case of instructional tasks, the student would typically only communicate a small fraction of the assumptions he or she used in the analysis. Such partial explanations must be filled in, to ensure that consistent and plausible solutions exist. We call this particular task *skeptical analysis* [6]. We are currently working on an account of this kind of reasoning, but have not progressed very far yet. Consequently, the rest of this section is very preliminary.

Skeptical analysis involves abductive inference. That is, given a description and partial explanation of how a system is supposed to behave, part of the task is finding a set of assumptions which would allow the behavior to deductively follow from them, the physical structure of the system, and physical laws (as expressed in domain theories). If no such set of assumptions exists, then the input explanation must be wrong, and a good skeptic will provide an explanation of what aspects lead to this abductive unsatisfiability. If such assumptions can be found, the next task of a skeptic is to search for strong arguments against them. If all the plausible sets of assumptions can be defeated, then again the explanation fails. If one set of assumptions survives, then the output of the skeptic should be the full explanation. If more than one set of assumptions survives, then the output of the skeptic should be a comparison of the alternatives, and suggestions about how one might distinguish between them.



The water falling from spigot (a) spins the water wheel (b), which turns the generator (c). The electricity produced by the generator is fed back into a pump (d), which pumps the water back up into the reservoir (e), to begin the cycle again. The excess electricity is tapped off (f) to supply the village with power.

Figure 3: What is wrong with this design?

An example will make this more concrete. Accounts of perpetual motion machines are a classic headache of patent examiners and debunkers of con men. Typically such schemes can be understood in terms of errors in the relative magnitudes of forces or in the neglecting of important factors, like friction. Consider the perpetual motion machine described in Figure 3. Clearly this explanation is incorrect, but why? The problem is that the amount of work required to raise the water is the same as that gained by letting it fall. Therefore even if the water wheel and pump and every other part of the system were absolutely perfect, there would not be any power left over. And since there is invariably friction in the water wheel, belt, pipes, pump, etc., the system will quickly come to a halt without some external source of power. Historically, it took quite a long time for the notion of power generation by perpetual motion to be laid to rest. While even qualitative versions of the law of energy conservation suffice to rule it out, formulating that perspective on systems took many centuries. Even today, students learning thermodynamics often look for loopholes, and cranks and con artists sometimes claim to find them.

We suspect that three common sources of errors in explanations are

1. *Invalid modeling assumptions.* Assuming that the strength of a container is effectively infinite, for example, can lead a designer to ignore the possibility of boiler explosions. Compositional modeling makes such assumptions explicit, and thus should support skeptical analysis.

2. *Invalid closed-world assumptions.* Ignoring physical processes implied by the description of the situation (e.g., friction) is a common feature of explanations of perpetual motion machines. Placing a notebook CPU next to its battery can provide a heat path that leads to the heat generated by the CPU ruining the battery. Again, domain theories created using compositional modeling are needed for such reasoning, because the conditions under which models are applicable must be explicitly represented and reasoned about.

3. *Ignoring operational constraints.* An engine design which requires bearings to be replaced every few hours is not likely to be very satisfactory. Such constraints will require domain theories that include more detailed, quantitative models.

The structure of programs that perform skeptical analysis depends in part on the task to be performed. For instance, a designer's assistant should note which aspects of a design need further refinement or analysis to confirm that the desired functionality is achievable. Furthermore, such an assistant should, given a set of domain-specific safety criteria, look for potential violations of these criteria. For instance, in designing a boiler one should think about how much time the operators will have between detecting a low level of feedwater before the boiler runs dry. If that time is too short for the crew to take effective action, the design should be changed to make the interval longer. (Qualitative reasoning suffices to point out the potential problem, but clearly quantitative analyses are needed to calculate intervals for both the dynamics and likely time to execute actions.) A skeptic which is part of a learning environment should gear its analysis of a student's explanation to the current pedagogical goals of the system.

5. Discussion

We have described our work in progress towards a qualitative account of representing and reasoning about function. To summarize our ideas so far:

- *Qualitative representations are central in reasoning about function.* Qualitative representations make explicit the nature of conceptual entities in the system and the causal connections between them. Qualitative representations provide part of the vocabulary needed for conceptual design, and provide a good starting point for students trying to learn a complex system.

- *Quantitative representations are useful for reasoning about function.* Often representations of function are conceived as more abstract than representations of physical behavior or structure. As argued above, indeed they often are. However, just as reasoning about structure and behavior benefits from using multiple levels of abstraction, so does reasoning about function. For example, numerical functional models can support reasoning about detailed behaviors without incurring the cost of detailed physical modeling.

- *Modeling assumptions and closed-world assumptions are a key part of the vocabulary connecting function to behavior.* In critiquing designs, the assumptions underlying the explanation of proposed behavior must be carefully checked for validity and completeness.

Acknowledgments

The port of the Feedback MiniLab to Windows in C++ was carried out by Scott Bucholtz. The development of the teleological vocabulary is being carried out in collaboration with Ron Ferguson and Seungseok Hyun. This research is supported by the Computer Science Division of the Office of Naval Research.

REFERENCES

1. Sembugamoorthy, V. and Chandrasekaran, B. ``Functional representation of devices and compilation of diagnostic problem-solving systems.", in Kolodner, J. and Riesbeck, C. (Eds.) *Experience, Memory, and Reasoning*, Erlbaum, 1986.

2. de Kleer, J. ``How Circuits Work", *Artificial Intelligence*, **24**, 1984.

3. Franke, D. ``Deriving and using descriptions of purpose," *IEEE Expert*, April, 1991, pp 44-47.

4. Vescovi, M. Iwasaki, Y., Fikes, R., and Chandrasekaran, B. ``CFRL: A language for specifying the causal functionality of engineered devices," to appear in *Proceedings of AAAI93*.

5. Falkenhainer, B. and Forbus, K. ``Compositional Modeling: Finding the Right Model for the Job", *Artificial Intelligence*, **51** (1-3), October, 1991.

6. Forbus, K. ``Qualitative process theory," *Artificial Intelligence*, **24**, 1984.

7. Forbus, K. ``The Logic of Occurrence", *Proceedings of IJCAI-87*, August, 1987.

8. Kuipers, B. ``Qualitative Simulation", *Artificial Intelligence*, **29**, September, 1986.

9. Forbus, K. ``An interactive laboratory for teaching control system concepts", BBN Technical Report No. 5511, January, 1984.

10. Forbus, K. and Falkenhainer, B. ``Self-explanatory simulations: An integration of qualitative and quantitative knowledge", *Proceedings of AAAI-90*, August, 1990.

11. Forbus, K. and Falkenhainer, B. ``Self-Explanatory Simulations: Scaling up to large models", *Proceedings of AAAI-92*, July, 1992.