Operator Procedure Verification with a Rapidly Reconfigurable Simulator

Yumi Iwasaki*, Robert Engelmore, Gary Fehr, Richard Fikes
Adam Farquhar, Thomas Gruber

Knowledge Systems Laboratory, Department of Computer Science
Stanford University
701 Welch Road, Palo Alto, CA
94304 USA
tel. (415)723-3444 fax (415)725-5850
* iwasaki@ksl.stanford.edu

KEY WORDS AND PHRASES

Automatic modeling, hybrid simulation, reconfigurable simulator, procedure verification

INTRODUCTION

Generating and testing procedures for controlling spacecraft subsystems composed of electro-mechanical and computationally realized elements has become a very difficult task. Before a spacecraft can be flown, mission controllers must envision a great variety of situations the flight crew may encounter during a mission and carefully construct procedures for operating the spacecraft in each possible situation. If, despite extensive pre-compilation of control procedures, an unforeseen situation arises during a mission, the mission controller must generate a new procedure for the flight crew in a limited amount of time. In such situations, the mission controller cannot systematically consider and test alternative procedures against models of the system being controlled, because the available simulator is too large and complex to reconfigure, run, and analyze quickly. A rapidly reconfigurable simulation environment that can execute a control procedure and show its effects on system behavior would greatly facilitate generation and testing of control procedures both before and during a mission.

There are several requirements that must be met by such a simulation system:

- **Reconfigurability** -- During a mission, the state of a component may change due to a fault or an unforeseen external event. During the design process, changes in the design of a physical system, which may occur concurrently with the design of an operating procedure, may require a modification to the procedure. For these reasons, it must be easy to change the simulation model to reflect the variety of configurations and conditions under which the spacecraft will be operated.
- **Simulation with imprecise or incomplete information** -- Exact and complete numerical data about the state of the system may not be available during design or in the presence of a fault. For example, when a leak is detected, the exact size of the leak is unlikely to be known. Therefore, the simulator must be able to predict behavior even if precise quantitative information about the state of the system is not available. If it is not possible to predict the behavior unambiguously, it should at least be able to produce a range of possible behaviors.
- **Explanation** -- When procedures produce unexpected results, it is difficult to interpret the raw simulation data, which may consist of values of hundreds of state variables in each of many states. The simulator should be able to produce a high-level, causal explanation of the simulation results, summarizing the salient information for the user and for documentation.

The How Things Work project at Stanford University has developed a system called DME (Device Modeling Environment) for modeling and simulating the behavior of electro-mechanical devices [1]. DME was designed to facilitate model formulation and behavior simulation of device behavior including both continuous and discrete phenomena. We are currently extending DME for use in testing operator procedures, and we have built a

knowledge base for modeling the Reaction Control System (RCS) of the space shuttle as a testbed. We believe that DME can facilitate design of operator procedures by providing mission controllers with a simulation environment that meets all these requirements.

DME: THE RAPIDLY RECONFIGURABLE MODELING AND SIMULATION SYSTEM

DME is an evolving prototype of a "designer's associate" system, intended to support the design of electro-mechanical devices by providing effective tools for simulating and analyzing the behavior of such devices [2]. The DME system is intended as an experimental testbed and foundation on which to build new representation and reasoning capabilities. DME has already been developed to a sufficient level of maturity to provide both a demonstration vehicle and a useful experimental testbed within the project. Currently, DME provides the following capabilities:

Model formulation: DME uses the given information about the structure of a device to generate a mathematical model of its behavior. DME has knowledge of the physical phenomena in the domain, represented as *model fragments* in CML [3], a compositional modeling language developed jointly by leading members of the qualitative reasoning research community. Each model fragment describes a particular aspect of a conceptually distinct physical phenomenon in terms of the conditions under which it occurs and the consequences of its occurrence.

Given the structure of a device in terms of its components and their connections along with the conditions that hold in an initial state, DME formulates a mathematical model of the behavior of the device by composing applicable model fragments and simulates the behavior. We have also been developing techniques for automatically formulating a simulation model that embodies the abstractions, approximations, assumptions, and perspectives that are appropriate for a given analysis task [4].

Simulation: DME uses the model it generates to perform behavior simulation. When sufficient numerical information is available, simulation is carried out numerically. Otherwise, it simulates behavior qualitatively. In both cases, DME can simulate a mixture of continuous and discrete phenomena.

Explanation: On the basis of an initial device model and the behavioral predictions obtained through simulation, DME can answer a range of user queries about the structure and behavior of the modeled system [5]. An important element of the explanation approach in DME is the use of the simulator's models, rather than ad hoc "causal models" that are built specifically for explanation generation. In explaining how things work, people do use causal terminology. However, when analyzing the behavior of devices, engineers use formalisms such as logical and mathematical constraints that are not causal. DME infers causal dependencies among modeled parameters by analyzing logical and mathematical constraints.

Reasoning about functions: Understanding how a device works requires knowledge of both its intended function and its actual behavior. DME provides a representation formalism, called CFRL, for specifying intended functionality and a verification mechanism to determine whether a simulated behavior achieves an intended function [6].

USE OF DME FOR OPERATOR PROCEDURE VERIFICATION IN THE RCS

We have built a DME knowledge base for modeling the Reaction Control System (RCS) of the space shuttle, and we are extending DME to do simulation and evaluation of operator procedures. The RCS is the system of thrusters that are used to control the attitude of the space shuttle while it is in orbit. Oxygen and fuel are fed to the RCS jets from separate tanks. The thrusters do not have pumps; instead the flow is maintained by keeping the tanks pressurized with helium. Each tank has a dedicated helium supply tank to maintain pressurization.

Mission controllers have carefully constructed procedures for operating the RCS under a variety of conditions. For instance, if a leak in the RCS is detected, then two procedures are employed to secure the system and identify the location of the leak. In order to secure the system, the astronaut must close all of the RCS

valves. The *RCS secure procedure* is to first close the valves nearest the thrusters and then to proceed upstream toward the helium tank until all of the valves have been closed. Once the system has been secured, the *isolation procedure* is to check the pressure in each of the segments between the closed valves. If the pressure in a particular segment is decreasing, then the leak has been isolated to that segment.

Even with procedures that seem simple, it is difficult to foresee the resulting interactions with the physical system. For instance, consider an alternative RCS secure procedure in which valves are closed in the opposite direction, starting with the main valve closest to the helium tank proceeding downstream towards the thrusters. Such a procedure is preferable for many systems -- as soon as the first (main) valve is closed, further propellant loss is prevented. In the RCS, however, this alternate procedure will result in cavitation inside the thrusters, leading to catastrophic damage.

Therefore, it is necessary to systematically test control procedures against models of the physical systems. When the execution of the procedure is simulated, the results need to be evaluated against the expected outcome of the procedure. At the time of this writing, DME has successfully formulated a behavior model of the RCS and simulated its behavior, given the specification of the RCS structure and initial conditions for the simulation. During simulation, DME allows the user to insert faults, such as leaks, or perform operator actions, such as opening and closing valves, to influence the course of behavior. As soon as any such changes are made, DME reformulates the model and continues simulating with the updated model. In this manner, DME has successfully predicted the results of the correct and incorrect valve closing sequences as described above in the presence of a leak.

We are currently extending DME in the following ways to enhance its support for procedure testing:

1) Develop the formal semantics of hybrid continuous and discrete models. This work is being carried out in collaboration with a team from the Xerox Palo Alto Research Center.
2) Extend the simulation mechanism to execute procedures automatically during simulation.
3) Expand CFRL to represent operator procedures and the intended effects of the procedures, which may not be explicit in the specification of the procedure itself.
4) Extend the verification mechanism to use the CFRL representation of operator procedures to verify whether the intended functions of a procedure are achieved in any given simulated trajectory of the system behavior.

An important type of knowledge about engineered devices is knowledge of its intended functions. Similarly, an important part of knowledge about operator procedures is knowledge of the function of the procedure, in other words, what the procedure is supposed to accomplish and how. CFRL was originally developed to represent device functions, but we believe it is also suitable for representing functions of operator procedures.

Figure 1 shows part of the proposed CFRL representation of the operator procedure to be invoked when over-pressurization of a propellant tank ($tk) is detected with both of the pressure regulators ($rega and $regb) open. Following the detection of the condition (node n0), the operator is to close the valves ($va and $vb) of both regulators (n1) and to open the thruster (n2), causing a decrease in the tank pressure (n3). When the pressure drops below 300 psi (n4), the operator is to reopen the valve of regulator A (n5). If the failure of regulator A is not detected by some other procedure (n7) within 60 seconds (n6), the operator is to conclude it is regulator B that has failed (n8).

The importance of functional knowledge extends not only to physical devices but also to *virtual devices* such as operator procedures. In the context of heterogeneous systems composed of electro-mechanical devices and control elements including digital computers and humans, operator procedures are as much a part of the system as any other physical component. It is important to evaluate the procedures under a variety of conditions, and such evaluation requires knowledge of their intended functions. We believe DME can facilitate the design of operator procedures by providing a means to explicitly represent a mission controller's intentions underlying a procedure and a useful simulation environment to evaluate whether a procedure achieves those intentions.

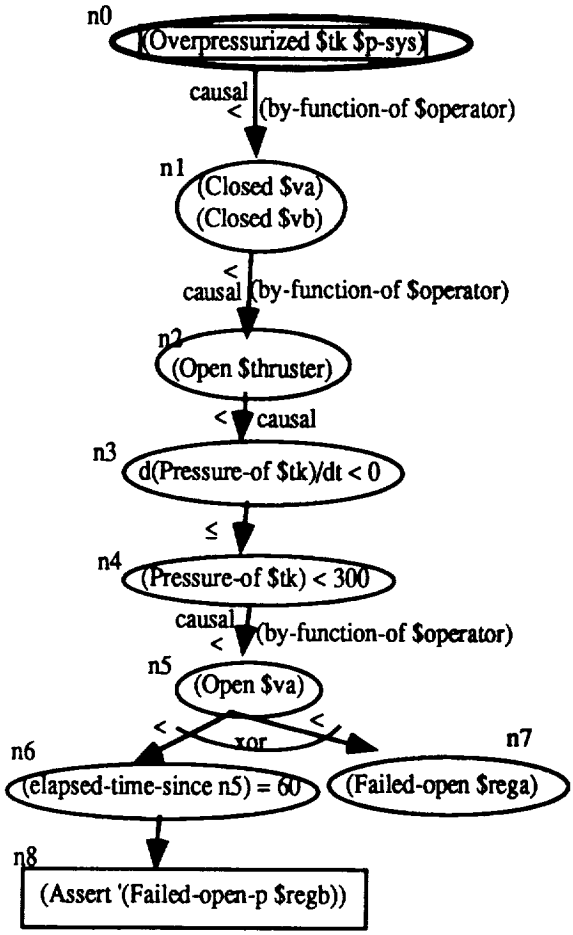

Figure 1. CFRL representation of an operator procedure

SUMMARY

In order to facilitate generation of procedures for operating complex dynamic spacecraft subsystems in a variety of expected and unexpected situations, it is essential to provide a modeling and simulation mechanism that can be quickly tailored to reflect a new configuration of the system being modeled. DME allows the user to change the system specification easily by altering the design or inserting faults to reflect a new situation. Reconfigurability of DME models comes from using compositional modeling technology. DME generates a new simulation model based on the altered specification and simulates the operator actions to predict the system behavior resulting from the actions. Such a facility will not only allow mission controllers to verify the safety of new procedures quickly, thereby avoiding unforeseen negative side effects, but also will be an essential component in a future automatic procedure generation and testing system.

ACKNOWLEDGMENT

This research was sponsored by the Advanced Research Projects Agency, ARPA Order 8607, monitored by NASA Ames Research Center under grant NAG 2-581; and by NASA Ames Research Center under grant NCC 2-537. We thank Drs. Vijay Saraswat and Daniel Bobrow of the Xerox Palo Alto Research Center for their involvement with us in the development of knowledge representation and formal semantics of hybrid systems.

REFERENCES

[1] Low, C.M. and Iwasaki, Y., *Device modelling environment: an interactive environment for modelling device behaviour.* Intelligent Systems Engineering, 1993. **1**(2): p. 115-145.

[2] Iwasaki, Y., *et al.* How Things are Intended to Work: Capturing Functional Knowledge in Device Design. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*. 1993. Chambery, France.

[3] Falkenhainer, B, *et al.* CML: A Compositional Modeling Language. KSL Technical Report, January 1994.

[4] Iwasaki, Y. and Levy, A.Y. Automated Model Selection for Simulation. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*. 1994. Seattle, WA. AAAI Press.

[5] Gautier, P.O. and Gruber, T.R. Generating Explanations of Device Behavior Using Compositional Modeling and Causal Ordering. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*. 1993. Washington, D. C. AAAI Press/the MIT Press.

[6] Vescovi, M., *et al.* CFRL: A Language for Specifying the Causal Functionality of Engineered Devices. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*. 1993. Washington, D. C. AAAI Press/The MIT Press.