From: AAAI Technical Report FS-92-03. Copyright © 1992, AAAI (www.aaai.org). All rights reserved.

The Stanford How Things Work Project*

Richard Fikes and Thomas Gruber and Yumi Iwasaki
Knowledge Systems Laboratory
Stanford University
701 Welch Road, Palo Alto, CA 94304

Abstract

We provide an overview of the Stanford How Things Work (HTW) project, an ongoing integrated collection of research activities in the Knowledge Systems Laboratory at Stanford University. The project is developing technology for representing knowledge about engineered devices in a form that enables the knowledge to be used in multiple systems for multiple reasoning tasks, and reasoning methods that enable the represented knowledge to be effectively applied to the performance of the core engineering task of simulating and analyzing device behavior. The central new capabilities currently being developed in the project are automated assistance with model formulation and with verification that a design for an electromechanical device satisfies its functional specification.

1 Introduction

The Stanford How Things Work Project is an ongoing integrated collection of research activities in the Knowledge Systems Laboratory at Stanford University [Fikes *et al.*, 1991] led by Prof. Richard Fikes. The overall objective of the project is to develop knowledge-based technology that will enable computer systems to offer intellectual assistance at high levels of competence to problem solvers and decision makers in all stages of the life cycle of engineered products. To achieve that objective, we are developing:

- Technology for representing knowledge about engineered devices in a form that enables the knowledge to be used in multiple systems for multiple reasoning tasks, and
- Reasoning methods that enable the represented knowledge to be effectively applied to the performance of the core engineering task of simulating and analyzing device behavior.

The knowledge to be represented includes a broad range of subjects, from the fundamentals of physics and engineering, to device models that describe device structure, behavior, and function, to the rationale for the design of specific devices. In order to directly support the reuse of encoded engineering knowledge bases, we are working with other research groups to establish a common device modeling language and a clearing house for device models. The common language will be based on and make use of the languages and tools being developed in the DARPA Knowledge-Sharing Initiative [Neches *et al.*, 1991].

The primary engineering task on which we are focusing is that of supporting the design of electromechanical devices by providing effective tools for *simulating and analyzing the behavior of such devices* in all stages of their design. Simulation technology has the potential of providing a rapid low-cost means of testing new designs for sophisticated equipment in many industries before acquisition decisions are made and expensive prototypes are built. In order to realize that potential, simulators need to have three key capabilities they are currently lacking. Namely, simulators need to be:

- **Applicable to partially specified designs** – Many of the financially significant decisions about new designs are made during conceptual and preliminary design stages. *Qualitative* simulation techniques are needed to obtain behavior analyses during those early stages of design, since the detailed design specifics required to do conventional numerical simulations are not yet available.
- **Rapidly reconfigurable** – Simulators need to be capable of supporting a broad range of tests of a new design that include variations in level of detail (from engineering analyses of individual subsystems to macro level mission effectiveness evaluations of the overall design), issues being addressed (fuel consumption rates, ease of operation, response speed, etc.), and operating conditions being considered (extreme weather, variations in operator training, etc.). No one simulator or one simulation model will be able to support such a range of tests. Thus, designers need to be provided with a *simulation foundry* that enables them to rapidly configure and run multiple simulations on an as-needed basis to answer specific analysis questions.
- **Self interpreting** – In order for designers to use simulators for multiple purposes on a routine basis, simulation results must be understandable with minimum effort. Thus, simulators need to provide significant assistance with the task of interpreting their output by producing summaries, explanations, and analyses which are directly oriented to a given analysis task.

We are developing knowledge-based technology that will remove those deficiencies. That is, we are developing augmentations to conventional simulators that will enable them to become applicable to partially specified designs, rapidly reconfigurable, and self-interpreting. In particular, we are developing techniques for:

- Automatically formulating a simulation model that embodies the abstractions, approximations, assump-

From AAAI Technical Report FS-92-03. Copyright © 1992, AAAI (www.aaai.org). All rights reserved.

tions, and perspectives that are appropriate for a given analysis task,

- Performing qualitative simulation of device modules which have not yet been designed in detail or whose detailed quantitative behavior is not relevant to a given analysis task,
- Automatically guiding a simulator to consider scenarios that are relevant to a given analysis task,
- Generating human-understandable *causal* explanations of simulation results,
- Automatically determining whether simulated behavior satisfies functional specifications, and
- Testing and automatically generating procedures for operating the device.

We are embodying the techniques developed in our research in an evolving prototype "designer's associate" system called the *Device Modeling Environment (DME)* [Iwasaki and Low, 1991]. The DME system is intended to be useful to the research community at large as an experimental testbed, educational tool, 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.

DME is intended to enable a designer to document a design as it evolves and to support experimentation with alternative designs. The current system is used as follows:

- **Designer describes device** – The designer selects components from a library and specifies the structural connections among the components.
- **Designer selects behavior models** – The designer selects from a library the models of component behavior that provide the abstractions, approximations, assumptions, and perspectives which are appropriate for the analysis he or she wants to do.
- **DME generates simulation model** – DME uses the device model specified by the designer to generate a qualitative *or* quantitative simulation model of the device.
- **Designer interactively guides the simulation** – The designer uses a simulator provided by DME to interactively explore possible device behaviors.
- **DME provides causal explanations of simulated behavior.**
- **Designer analyzes behavior** – The designer compares the predicted behavior with the intended functionality of the device.

2 New capabilities being produced

The current DME system embodies state of the art research results. It provides an integrated set of tools for performing what might be characterized as a limited form of semi-automatic behavior analysis. For example, the system automatically formulates a simulation model, but only after the designer has selected from the system's model library appropriate behavior models for each device component.

Our current research is focused on taking steps toward providing a designer with a *comprehensive* and *fully automated* behavior analysis of a device being designed. Our three year goal is to develop new capabilities and integrate them into DME so that the system could be used as follows:

- **Designer describes device** – In addition to the current facilities for selecting components from a library and specifying the structural connections among the components, new facilities will be developed to enable the designer to specify:
  - Intended functionality of a device,
  - Expected operator interactions with a device,
  - Assumptions about the environment in which a device will be operating, and
  - Rationale for design decisions;
- **DME formulates appropriate behavior model** – New facilities will be developed that will enable DME to determine the abstractions, approximations, assumptions, and perspectives that are appropriate for specific analysis tasks such as testing whether the device design satisfies the functional specifications.
- **DME generates appropriate simulation model** – DME will use the structural and behavioral device models to generate a simulation model of the device that *intermixes* qualitative and quantitative simulation as needed. New facilities will be developed to enable it to select an appropriate qualitative or quantitative simulator for each device module depending on the level of detail at which the module has been designed and the level of detail required by the analysis task.
- **DME guides the simulation** – New facilities will be developed to enable DME to direct the simulator to consider scenarios that are relevant to a given analysis goal such as testing whether the functional specification is satisfied.
- **DME determines whether behavior achieves the intended functionality** – New facilities will be developed to enable DME to compare the simulated behavior with the functional specification. In cases where the behavior does not satisfy the specifications, DME will be able to provide feedback in the form of additional constraints on the design which would guarantee that the device behaves as intended.
- **DME explains behavior analysis results** – In addition to the current facilities for providing causal explanations of simulated behavior, new facilities will be developed to explain how and why the design either does or does not satisfy the functional specification.

An additional significant capability being developed in our project which is not highlighted in the above scenario is the use of DME for designing and analyzing procedures for operating a device. For example, DME seems particularly useful for assisting with the verification of procedures that

From: AAAI Technical Report FS-92-03. Copyright © 1992, AAAI (www.aaai.org). All rights reserved.

respond to device malfunctions in that it enables simulation models to be rapidly reformulated to reflect maintenance actions and can explain the effects caused by the procedures. We are currently working with NASA on such a procedure verification application in which DME will be used for both procedure debugging and operator training.

The central new capabilities currently being developed for DME are automated assistance with model formulation and with verification that a design satisfies its functional specification. Our approach to achieving these capabilities is summarized in the sections below.

2.1 Automatic Model Formulation

We are developing methods for providing automated assistance with the core problem of model formulation – a service that will help engineers build nontrivial models of device behavior for specific purposes.

The state of the art in model formulation today is model configuration from libraries of component models. Simulators such as SPICE [Katzenelson, 1966] and those for VHDL [Harr and Stanculescu, 1989] are based on libraries of component models which have been preformulated by modeling experts. The user selects components and configures them, and then the system compiles the code necessary to run a simulation and plot the trajectories of variables.

Today's component-based model libraries are most successful in those domains where components are well-defined idealizations at a single level of abstraction, such as logical circuits. The mapping between physical components and idealized component models is simple, and there is exactly one behavior model associated with each component model. Thus, the engineer's part of the model formulation task is simplified in that he or she need only specify a component connection topology.

However, in most domains and tasks, the mapping from phenomena of interest in a physical system to a set of possible behavior models is complex and the result of nontrivial reasoning. In doing model formulation, an engineer must identify the *relevant abstractions* to model, deciding, for example, whether to treat the load of an electrical power supply as a single resistive element or as a system with components that vary in their power usage. The engineer must also make *simplifying assumptions* and *approximations*, such as to assume no friction in a gear train or that a valve can be modeled as a discrete switch. The engineer makes these modeling choices to produce a model which answers a particular information need in a reasonable amount of time.

The power of the library approach derives from the reuse of the component models and the automatic assembly of system models from partial descriptions. DME will achieve the same advantages of knowledge reuse and automation, but for a more general class of domains and for multiple modeling purposes.

Even in domains such as analog circuits where there is a large library of ready-made simulation modules for standard components, building a model of an entire system is not an easy task. There are typically many simulation modules for each type of component, each based on different simplifying assumptions and approximations which are not stated explicitly. Therefore, a significant amount of effort and expertise is required for engineers to use even off-the-shelf simulation modules to assemble a model of a whole system. Engineers often prefer to write their own modules instead of using off-the-shelf modules precisely because they do not know all the underlying assumptions and do not trust their results.

DME will enable *knowledge reuse* by providing the representation and architecture for model libraries containing a comprehensive body of behavior model fragments, each making particular abstractions and approximations and conditioned on *explicitly represented* modeling assumptions. The formalism and examples will allow engineers to fill the libraries with model fragments covering those phenomena they need to model. We expect that a market will develop for these models, possibly driving a small industry of component-model-building specialists (as in electronics).

DME will provide *automated model formulation assistance* using these libraries. The assistance will change the nature of the interaction between the human engineer and the computational environment. Instead of operating at the level of equations or fixed-level component models, the engineer may specify the high-level device structure, the simulation goal, the utility criteria, a description of the context of use, and any initial conditions. The system will take an active role in selecting appropriate model fragments to construct a complete and coherent simulation model. This advance in model formulation is similar to the improvement in software development from early assembly-language programming to Fourth Generation Language environments.

2.2 Automatic Behavior Verification

Simulating the behavior of a device is an important means of ascertaining whether a device achieves its intended function. In order to provide useful feedback for further development of a design, the system must be able to evaluate the predicted behavior with respect to knowledge of the intended function. Does the predicted behavior achieve the desired function? Does it do so in the way the designer intended? These are crucial questions in providing useful feedback to the designer. In order for a model-based reasoning system to answer such questions, it must have knowledge of the intended function of the device (i.e., *What* it is supposed to do.) and the expected behavior (i.e., *How* it is supposed to achieve the function).

In the HTW project, we are working on a technique for design verification using both knowledge of the structure of a device and its intended functions. In particular, we address the question of when one can say a behavior predicted by a prediction system achieves the desired function in the manner intended by the designer. For this purpose, we are extending the Functional Representation language [Sembugamoorthy and Chandrasekaran, 1986] to represent the function of a device and the expected causal process for achieving the function.

In contrast to our bottom-up approach to predicting behavior of the overall device from the knowledge of individual components and physical processes, we take a top-down approach in representing functions. We describe the function of the overall device first, and then we describe the behavior of each component in terms of how it contributes to the overall function. Verification that a behavior of a

From: AAAI Technical Report FS-92-03. Copyright © 1992, AAAI (www.aaai.org). All rights reserved.

designed artifact achieves the desired goal must ascertain the following:

- The overall function of the device is achieved,
- The expected chain of events happen in the predicted behavior, and
- The causal connections expected between events exist in the predicted behavior.

We are developing a procedure for verifying a behavior generated by DME based on a formal definition of matching between the predicted behavior and the description of a causal process to achieve a function [Iwasaki and Chandrasekaran, 1992].

We believe that evaluating a behavior with respect to the expected causal process as well as the function improves the chances of uncovering hidden flaws in a design that may otherwise go undetected at an early stage. Furthermore, knowing how the intended function of a device is accomplished is just as important a part of understanding how things work as knowing how its structure gives rise to its actual behavior. Such knowledge of the functional aspects of a device is crucial not only in design but also in diagnosis and monitoring. Our work on representing functions and defining what the representation really means with respect to behaviors is a critical step in making DME a truly useful tool in engineering problem solving.

References

[Fikes *et al.*, 1991] Richard Fikes, Tom Gruber, Yumi Iwasaki, Alon Levy, and Pandu Nayak. How things work project overview. Technical Report KSL 91-70, Stanford University, Knowledge Systems Laboratory, 1991.

[Harr and Stanculescu, 1989] R. Harr and A. Stanculescu, editors. *Applications of VHDL to Circuit Design.* Kluwer Academic Publishers, Norwell, Massachusetts, 1989.

[Iwasaki and Chandrasekaran, 1992] Y. Iwasaki and B. Chandrasekaran. Design verification through function and behavior-oriented representations: Bridging the gap between function and behavior. In *Proceedings of the Second International Conference On Artificial Intelligence in Design*, 1992.

[Iwasaki and Low, 1991] Yumi Iwasaki and Chee Meng Low. Model generation and simulation of device behavior with continuous and discrete changes: the details. Technical Report KSL 91-30, Stanford University, Knowledge Systems Laboratory, 1991.

[Katzenelson, 1966] J. Katzenelson. Aednet: A simulator for nonlinear networks. In *Proceedings of the IEEE*, 1966.

[Neches *et al.*, 1991] Robert Neches, Richard Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(3):37–56, 1991.

[Sembugamoorthy and Chandrasekaran, 1986] V. Sembugamoorthy and B. Chandrasekaran. Functional representation of devices and compilation of diagnostic problem-solving systems. In *Experience, Memory, and Reasoning.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986.