COMPUTER-AIDED OPERATIONS ENGINEERING WITH INTEGRATED MODELS OF SYSTEMS AND OPERATIONS

Jane T. Malin
NASA
Automation and Robotics Division
Johnson Space Center, ER2
Houston, TX 77058

Dan Ryan
MITRE
Johnson Space Center, ER2
Houston, TX 77058

Land Fleming
Lockheed Engineering & Sciences Co.
Houston, TX 77058

ABSTRACT

CONFIG 3 is a prototype software tool that supports integrated conceptual design evaluation from early in the product life cycle, by supporting isolated or integrated modeling, simulation, and analysis of the function, structure, behavior, failures and operation of system designs. Integration and reuse of models is supported in an object-oriented environment providing capabilities for graph analysis and discrete event simulation. Integration is supported among diverse modeling approaches (component view, configuration or flow path view, and procedure view) and diverse simulation and analysis approaches. Support is provided for integrated engineering in diverse design domains, including mechanical and electro-mechanical systems, distributed computer systems, and chemical processing and transport systems. CONFIG supports abstracted qualitative and symbolic modeling, for early conceptual design. System models are component structure models with operating modes, with embedded time-related behavior models. CONFIG supports failure modeling and modeling of state or configuration changes that result in dynamic changes in dependencies among components. Operations and procedure models are activity structure models that interact with system models. CONFIG is designed to support evaluation of system operability, diagnosability and fault tolerance, and analysis of the development of system effects of problems over time, including faults, failures, and procedural or environmental difficulties.

INTRODUCTION

The core of engineering design and evaluation focuses on analysis of physical design. Thus, today's computer-aided engineering software packages often do not provide enough support for conceptual design early in the life cycle or for engineering for operation, fault management, or supportability (reliability and maintainability). Benefits of engineering for operations and supportability include more robust systems that meet customer needs better and that are easier to operate, maintain and repair. Benefits of concurrent engineering include reduced costs and shortened time for system development. Integrated modeling and analysis of system function, structure, behavior, failures and operation is needed, early in the life cycle.

Conventional system modeling approaches are not well suited for evaluating conceptual designs early in the system life cycle. These modeling approaches require more knowledge of geometric or performance parameters than is usually available early in design. More abstracted models can support early conceptual design definition and evaluation, and also remain useful for some later analyses. Component-connection models provide one such useful abstraction,

and discrete events are another. Discrete event simulation technology combines both abstractions, for evaluation of conceptual designs of equipment configurations in operations research (3). CONFIG uses these abstractions, with some enhancements, to define and evaluate conceptual designs for several types of systems.

The initial CONFIG project goal was to support simulation studies for design of automated diagnostic software for new life-support systems (9). The problem was to design an "expert system" on-line troubleshooter before there was an expert. The design engineer could use a model of the system to support what-if analyses of failure propagation, interaction, observability and testability. This activity is similar to Failure Modes and Effects Analysis (5), but uses comparative simulations of failure effects to develop diagnostic software. Conventional simulation software was not up to this challenge, but discrete event simulation software has been. CONFIG supports the use of qualitative models for applying discrete event simulation to continuous systems.

A major design goal for CONFIG is to support conceptual design for operations and safety engineering. Major tasks in conceptual design are design definition, evaluation (by simulation and analysis) and documentation. Operations engineering focuses on the design of systems and procedures for operating, controlling and managing the system in normal or faulty conditions. Safety engineering focuses on prevention of hazardous effects and conditions in the physical system or its operation. In these types of engineering, complex interactions and interfaces among system components and operations must be a focus.

Another design goal of CONFIG is to bridge the gaps between physical design engineering and other types of engineering. Component-connection representations are well suited for modeling and defining physical system designs (as structures of interacting components) and operations designs (as structures of interacting actions), as well as interactions between system components and operational actions. Discrete event models have been used for this type of modeling for queueing and scheduling problems, but can be extended to support conceptual modeling in operations and safety engineering. This type of modeling is also compatible with systems engineering function diagrams (1).

CONFIG 3

The project approach has been to incrementally integrate advanced modeling and analysis technology with more conventional technology. The prototype integrates qualitative modeling, discrete event simulation and directed graph analysis technologies for use in analyzing normal and faulty behaviors of dynamic systems and their operations. The prototype has been designed for modularity, portability and extensibility. A generic directed graph element design has been used to standardize model element designs and to promote extensibility. This directed graph framework supports integration of levels of modeling abstraction and integration of alternative types of model elements.

Enhanced Discrete Event Simulation Capabilities

In traditional discrete event modeling and simulation, state changes in a system's entities, "events", occur discretely rather than continuously, at nonuniform intervals of time. Throughout simulation, new events are added to an event list that contains records of events and the times they are scheduled to occur. Simulation processing jumps from one event to the next, rather than occurring at a regular time interval. Computation that results in creation of new events is localized in components, which are connected in a network. Any simulation run produces a particular history of the states of the system. Statistical simulation experiments, using random variables in repeated simulation runs, are used to compare design alternatives.

To enhance this discrete event simulation approach to accommodate continuous systems, a number of new concepts and methods were developed. These include component models with operating modes, types of links connecting components ("relations" and "variable clusters"), state transition structures ("processes"), methods for representing qualitative and quantitative functions ("process language"), and a new simulation control approach.

Digraph Analysis Capabilities

The CONFIG Digraph Analyzer (DGA) makes graph analysis techniques available for evaluating conceptual designs of systems and their operations. The DGA is based on reachability search, and is implemented generically for application to the many types of graph data structures in CONFIG. DGA can support analyses of completeness, consistency and modularity. Analysis of failure sources and impacts can be done by tracing the paths from a given failure.

System Modeling

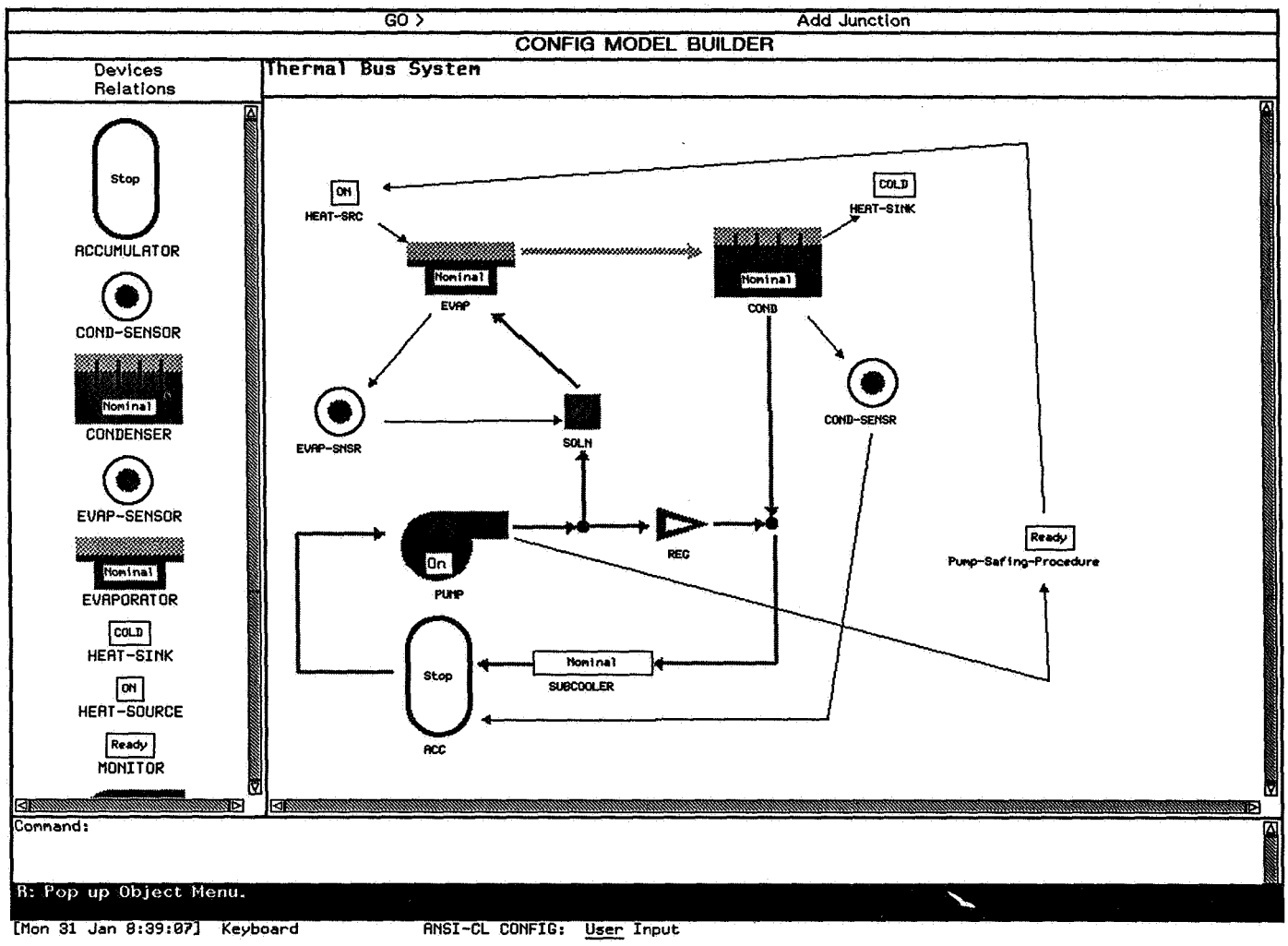
Devices are the basic components of a CONFIG system model. Relations are basic connectors for building topological model structures with devices. Device behavior is defined within operating and failure Modes, which contain mode dependent and mode transition Processes. Modes are connected together in a mode transition digraph which delineates the transition dependencies among the individual modes. A Thermal Bus System model, shown in the Figure, illustrates a CONFIG model of a system of connected devices (heat exchangers, valves, sensors, accumulator, etc.) for removing heat. The behavior of each device depends on its current mode and on changes in its input received from other devices via relations or via the global flow path manager.

Device Processes define changes in device variables as events with time delays, which are conditionally invoked and executed during a simulation. Processes define time-related behavioral effects of changes in device input variables, both direction of change and the new discrete value that will be reached, possibly after a delay. Faults and failures can be modeled in two distinctly different ways. Failure modes can be used to model device faults. Mode-transition processes can be used to model latent failures or triggering failures that prevent or cause mode changes. Relations connect devices via their variables, so that state changes can propagate along these relations during simulations. Related variables are organized into variable clusters, to separate types of relations by domain (e.g., electrical vs. fluid connections). Relations can also be used to make connections between system Devices and Activities in operations models.

Flow Path Modeling

Flow is a property of many systems, whether the substance flowing is a liquid or information. There are two difficulties in modeling flows with local device processes. First, flow is a global property of the topology of the modeled system and the substances flowing within it. Second, while dynamic changes in system structure and flow can occur during operations, process descriptions involving flow must often rely on assumptions of static system topology. These factors would limit the reusability of device descriptions to a limited set of system structures.

A flow-path management module (FPMM) has been implemented to address these problems. The FPMM is separate from the module implementing local device behavior, but the two modules are interfaced via flow-related state variables in the devices. When FPMM is notified during simulation of a local change in device state, it recomputes the global effects on flows produced by the local state change. The FPMM then updates the state of flow in all affected


GO >
Add Junction
CONFIG MODEL BUILDER
Devices
Relations
Thermal Bus System
Stop
ACCUMULATOR
COND-SENSOR
Nominal
CONDENSER
EVAP-SENSOR
Nominal
EVAPORATOR
COLD
HEAT-SINK
ON
HEAT-SOURCE
Ready
MONITOR
ON
HEAT-SRC
COLD
HEAT-SINK
Nominal
EVAP
Nominal
COND
EVAP-SNSR
SOLN
COND-SENSR
On
PUMP
REG
Ready
Pump-Safing-Procedure
Stop
ACC
Nominal
SUBCOOLER
Command:
R: Pop up Object Menu.
[Mon 31 Jan 8:39:07] Keyboard
ANSI-CL CONFIG: User Input

devices. This design permits the user to write reusable local device process descriptions that do not depend on any assumptions concerning the system topology.

FPMM uses a simplified representation of the system, as a collection of aggregate objects, or "circuits." Further abstraction is achieved by identifying serial and parallel clusters in the circuit (6). In many cases, configuration determination alone is sufficient to verify flow/effort path designs, or to establish flow paths for a continuous simulation, for reconfiguration planning, and for troubleshooting analysis (see Ref. [2] on cluster-based design of procedures for diagnosis, test, repair and work-around in a system).

Operations modeling

Activities are the basic components of a CONFIG operations model, and are connected together in action structures with Relations. They represent procedures or protocols that interact with the system, to control and use it to achieve goals or functions. Each activity model can include specifications for what it is intended to achieve or maintain. Activity behavior is controlled in a sequence of phases, ending in an evaluation of results. Activity behavior is defined by processes that model direct effects of actions, or that control device operation and mode transitions to achieve activity goals. Relations define sequencing and control between activities and connect Devices with device-controlling Activities.

Operations models are designed to support operation analysis with procedure models. These models are designed to support analysis of plans and procedures for nominal or off-nominal operation. The procedure modeling elements are designed for reuse by intelligent replanning software, and for compatibility with functional modeling in systems engineering.

Model Development and Integration Capabilities and Approach

CONFIG provides intelligent automation to support nonprogrammer and nonspecialist use and understanding. CONFIG embeds object-oriented model libraries in an easy-to-use toolkit with interactive graphics and automatic programming.

CONFIG provides extensive support for three separable yet tightly integrated phases of user operation during a modeling session: Library Design, Model Building, and Simulation and Analysis. This includes a graphical user interface for automated support of modeling during each of the phases including the development of object-oriented library element classes or templates, the construction of models from these library items, model inspection and verification, and running simulations and analyses.

The integration between the phases enables an incremental approach to the modeling process. Lessons learned from analyses can be repeatedly and rapidly incorporated by the user into an initially simple model. Support for these phases as separate user activities fosters the achievement of concurrent engineering goals. Different users can define library elements, build models, and analyze models at different times depending on area of expertise and availability of resources. Support for the model building phases spans all types of modeling that can be performed in CONFIG including component structure, behavior and flow, and activity goals and structure.

Hosting

CONFIG is implemented in software that is portable to most Unix work stations. The Common LISP Object System (CLOS) is a highly standardized language, with compilers for most of the commonly available work stations. The user interface was implemented using the Common LISP Interface Manager (CLIM), another standardized tool built on CLOS.

CONCLUSIONS

CONFIG is designed to model many types of systems in which discrete and continuous processes occur. The CONFIG 2 prototype was used to model and analyze: 1) a simple two-phase thermal control system based on a Space Station prototype thermal bus, 2) a reconfigurable computer network with alternate communications protocols, and 3) Space Shuttle Remote Manipulator System latching and deployment subsystems (7). The core ideas of CONFIG have been patented (8). CONFIG 3 has added capabilities for graph analysis and for modeling operations and procedures.

The CONFIG prototype demonstrates advanced integrated modeling, simulation and analysis to support integrated and coordinated engineering. CONFIG supports qualitative and symbolic modeling, for early conceptual design. System models are component structure models with operating modes, with embedded time-related behavior models. CONFIG supports failure modeling and modeling of state or configuration changes that result in dynamic changes in dependencies among components. Operations and procedure models are activity structure models that interact with system models. The models support simulation and analysis both of monitoring and diagnosis systems and of operation itself. CONFIG is designed to support evaluation of system operability, diagnosability and fault tolerance, and analysis of the development of system effects of problems over time, including faults, failures, and procedural or environmental difficulties.

ACKNOWLEDGEMENTS

The authors wish to thank Bryan Basham, Leslie Ambrose, Ralph Krog, Debra Schreckenghost, Brian Cox, Daniel Leifker and Sherry Land for their contributions to CONFIG design and implementation.

REFERENCES

1. Alford, M., "Strengthening the System Engineering Process," ENGINEERING MANAGEMENT JOURNAL, Vol. 4, No. 1, March, 1992, pp. 7-14.

2. Farley, A. M., "Cluster-based Representation of Hydraulic Systems," PROC. 4TH CONFERENCE ON AI APPLICATIONS, March, 1988, pp. 358-364.

3. Fishman, G. S., PRINCIPLES OF DISCRETE EVENT SIMULATION, Wiley, New York, NY, 1978.

4. Forbus, K., "Qualitative Physics: Past, Present, and Future," EXPLORING ARTIFICIAL INTELLIGENCE, Eds. H. Shrobe and AAAI, Morgan Kaufmann, San Mateo, CA, 1988.

5. Fullwood, R. R. and Hall, R. E., PROBABILISTIC RISK ASSESSMENT IN THE NUCLEAR POWER INDUSTRY: FUNDAMENTALS AND APPLICATIONS, Pergamon Press, 1988.

6. Liu, Z. and Farley, A. M., "Structural Aggregation in Common-Sense Reasoning," PROC. 9TH NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (AAAI-91), AAAI Press, Menlo Park, CA, July, 1991, pp. 868-873.

7. Malin, J. T., Basham, B. D., and Harris, R. A., "Use of Qualitative Models in Discrete Event Simulation for Analysis of Malfunctions in Continuous Processing Systems," ARTIFICIAL INTELLIGENCE IN PROCESS ENGINEERING, Ed. M. Mavrovouniotis, Academic Press, 1990, pp. 37-79.

8. Malin et al., U. S. PATENT 4,965,743, "Discrete Event Simulation Tool for Analysis of Qualitative Models of Continuous Processing Systems," October, 1990.

9. Malin, J. T., and Lance, N., "Processes in Construction of Failure Management Expert Systems from Device Design Information," IEEE TRANS. ON SYSTEMS, MAN, AND CYBERNETICS, Vol. SMC-17, 1987, pp. 956-967.

10. Malin, J. T. and Leifker, D. B., "Functional Modeling with Goal-Oriented Activities for Analysis of Effects of Failures on Functions and Operations," INFORMATICS AND TELEMATICS, Vol 8, No. 4, 1991, pp. 353-364.