N89-19831

A DESIGN AND IMPLEMENTATION METHODOLOGY FOR DIAGNOSTIC SYSTEMS

Linda J. F. Williams
Mcdonnell Douglas
16055 Space Center Blvd.
Houston, TX 77062

ABSTRACT

This paper presents a methodology for design and implementation of diagnostic systems. Also discussed are the advantages of embedding a diagnostic system in a host system environment. The methodology utilizes an architecture for diagnostic system development that is hierarchical and makes use of object-oriented representation techniques. Additionally, qualitative models are used to describe the host system components and their behavior. The methodology architecture includes a diagnostic engine that utilizes a combination of heuristic knowledge, causal knowledge and system structure knowledge to control the sequence of diagnostic reasoning.

The methodology provides an integrated approach to development of diagnostic system requirements that is more rigorous than standard systems engineering techniques. The advantages of using this methodology during various lifecycle phases of the host systems (e.g. National Aerospace Plane (NASP)) include: the capability to analyze diagnostic instrumentation requirements during the host system design phase, a ready software architecture for implementation of diagnostics in the host system, and the opportunity to analyze instrumentation for failure coverage in safety critical host system operations.

1.0 INTRODUCTION

Space transportation systems are among the most advanced and complex systems being designed today and provide a wide variety of physical systems that will require some measure of intelligent automation. The increased complexity of these types of systems has led to extremely complex operations in manned systems that will continue to overburden flightcrews. There is an inherent need to assist crew personnel in complex system operations through intelligent automation. In the case of unmanned systems, intelligent automation integrated at the design phase will greatly assist in the selection of optimal instrumentation sets. (By optimal instrumentation set we are referring to the set of instrumentation that will provide the most complete information during monitoring, control and fault diagnosis functions).

An area that can be greatly enhanced by intelligent automation is Fault Diagnosis, Fault Isolation and Recovery (FDIR) functions. Intelligent FDIR automation will allow more productive and effective use of crew personnel, fewer system shutdowns, reduced system downtime, improved safety, maintainability, reliability, and reduce crew cognitive overload.

The need for intelligent Fault Diagnosis, Fault Isolation and Recovery (FDIR) automation in several different physical systems has lead to design concepts that are being developed into a methodology and implemented in a generic software tool that will provide automated FDIR functions for any physical system. This generic software tool is the System Diagnostic Engine (SDE). The objective of the SDE project is to define and develop a methodology for design, analysis and implementation of diagnostic systems. The goal of the SDE project is to develop this methodology into a generic, domain independent software tool that will provide automated FDIR capabilities at reasonable execution speeds with database integration and knowledge acquisition capabilities.

The use of a generic software tool like the SDE will augment heuristic-based (rule-based) automation by reducing the brittleness that is inherent in heuristic-based systems. The SDE also has the potential for reducing the initial cost of the system design and automation of the FDIR procedures.

2.0 INTELLIGENT FDIR AUTOMATION

Growing interest in automating FDIR functions is evident in several projects throughout the aerospace community including but not limited to: Faultfinder (1), an intelligent aid for assisting flight crew in FDIR functions; Helix (7), intelligent aide for diagnosing the power train of twin-engine helicopters; and Muxpert (4), intelligent aide for diagnosing AH-64A Apache multiplex subsystems. Each project deals with a different domain but use similar architectures and qualitative reasoning techniques to improve automated FDIR capability.

2.1 Qualitative Reasoning

Qualitative reasoning is an area of Artificial Intelligence (AI) research that addresses problems of reasoning about physical systems. This includes the areas of causal reasoning (9,10,3), reasoning from structural knowledge (9,10,3), qualitative modeling (3), qualitative simulation (8), etc. Qualitative reasoning shows great promise in augmenting the "traditional" expert system technology. (By "traditional" expert system technology we are referring to the technology of developing expert systems using heuristic knowledge). When people reason about physical systems they use more than just experiential (or heuristic) knowledge, they in fact use commonsense knowledge and often develop a mental model (with the appropriate level of detail necessary) to understand the physical system's behavior and to reason about novel faults. Qualitative reasoning research involves automating this human reasoning process. (3,10) A common theme of much of the qualitative reasoning research is explaining how physical systems work using a description of system structure and behavior. The behavioral description of the physical system can be derived from the system structure; structure being the physical system's components, the connectivity between the components, and the component behaviors. (9) The term 'behavior' refers to the observable changes (over time) of the state of the components and the system as a whole. Components have individual behaviors and the collective interactive component behavior results in the behavior of the system as a whole. While the 'structural description' consist of the individual variables that characterize the system and their interactions, the 'behavioral description' consist of the potential behaviors of the system. The 'functional description' of a physical system reveals the purpose of a structural component or connection in producing the behavior of a system. For example, the function of a release valve on a pressurized tank is to prevent an explosion; the behavior of the system as a whole is to maintain a pressure below a certain limit.(8) Reasoning about the functional description of a physical system can facilitate understanding of the system behavior. This can lead to interesting optimizations in system design and creative alternatives to fault recovery procedures for physical systems. A completely different component may be substituted for a piece of a larger system if the function of the two components are equal. For example, a light bulb could be used to replace (or partially replace) a small heat source. In a fault recovery situation, the location of the replacement component is an additional constraint that must be considered. (The light bulb must be in an appropriate location to be considered for use as a heat source.)

2.1.1 Causal Reasoning

Qualitative reasoning research supports the following reasoning tasks: 1) simulation - starting with a structural description of a physical system, and initial conditions, determine a likely course of future behavior (8); 2) envisionment - starting with a structural description, determine all possible behavioral sequences (6); 3) diagnosis - comparing composed behavior (as computed from a structural description) with specified desired behavior (5); 4) verification - ascertain that a particular implementation structure has a composite behavior which matches the desired behavior specification (2).

A common criteria for explanation in each of these qualitative reasoning tasks involves causal reasoning. Causal reasoning refers to the use of causal knowledge (i.e. cause and effect information) about a physical system to derive knowledge about the behavior and function of the physical system. This reasoning method contrast with standard physics where systems are described by differential equations which provide constraints on the dynamics of the system state variables. Although the analytical techniques are capable of capturing a more complete state of knowledge, people rarely use analytical techniques when reasoning about a physical system. More often people will use causal information in mental models to gain an understanding of system behavior. Since people are very capable of performing FDIR functions without solving differential equations in their heads, it is reasonable to assume that causal reasoning is useful in intelligent FDIR automation. In fact, a great deal of information can be derived from an understanding of component connectivity and causal processes that underly a physical system.

2.1.2 Connectivity Representation

The structural description of a physical system consist of system components, the connectivity between components, and

component behaviors (as discussed in section 2.1). The component connectivity in a physical system structure can be represented using "adjacency" and "reachability". Adjacency describes components that are directly connected to one another (e.g. a resistor is directly connected to a wire). Reachability refers to components that can have an effect on one another but are not adjacent. In figure 1 the regulator is adjacent to the hall-sensor and the hall-sensor is adjacent to the amp-trigger, but there is also a reachable connection (effect) from the regulator to the amp-trigger. There are inherent advantages to "gathering" adjacent and reachable connectivity information about a physical system into a computable structure. "Gathering" information refers to the automatic collection of data from the structural description. This can be accomplished by computing the connected reachability information from the adjacency data which was derived from the structural description.* By "computable structure" we are referring to a stable computer structure (e.g. a matrix) that is developed a single time at the initial execution stage and provides rapid access to data. Using this method allows information associated with data propagating through the components of the system to be readily available without simulating propagation (e.g. tracing through a frame or object type of representation). Examining the reachability data also gives us clues about the physical system's structural description that may not be obvious in a frame or object type representation. (Especially in system representations that contain a large number of interconnected components). An example of this type of undetected representation would be a circuit (as the term applies to graph theory). A circuit can be thought of as a continuous loop in a structure and would have significant impact on data propagating through a physical system. The ability to detect this type of structural information could assist in creating a more complete automated FDIR capability. Having access to the reachability data provides us with a relatively simple method for analyzing diagnostic instrumentation requirements during the physical system design phase. Understanding connectivity in a physical system allows us to better analyze the proficiency of a particular instrumentation set for handling failure coverage in safety critical operations.

Methods for combining the types of reasoning discussed above, and ways to apply qualitative reasoning techniques to permit combinations of qualitative reasoning tasks in automated diagnostic reasoning systems are being studied in the projects listed above (section 2.0) and in the System Diagnostic Engine (SDE) project described in the following paragraphs.

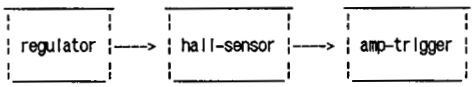


Figure 1 Adjacency/Reachability

3.0 SYSTEM DIAGNOSTIC ENGINE

The idea of a SDE was conceived from attempting to enhance the automated FDIR capability in a heuristic-based expert system by combining causal knowledge and experiential knowledge. (10) The approach to diagnosing a malfunctioning system through the use of a deep understanding of the fundamental structure and behavior of the system and its components has the target of providing an expert's troubleshooting ability without explicitly modeling the expert. The advantage of this approach are especially apparent when automating FDIR operations. Certain aspects of the FDIR operations will change when moving from a manned operational mode to a fully automated operational mode. A human operator may be required to make observations that are unavailable to an automated system. For example, a voltage reading from a meter might be necessary for fault diagnosis in a manned operational mode. An automated system cannot access the same information and must rely on other methods to derive the same results. Integrating intelligent FDIR automation during system design will assist in selecting the optimal set of instrumentation to allow the automated system to have access to appropriate sensor information.

3.1 SDE Architecture

The methodology architecture uses a combination of heuristics, qualitative models, 1st principles and causal information to reason about system structure, functions and associated faults. The methodology will support design of systems for diagnosability and intelligent automated control for FDIR, and will augment heuristic (rule-based) expert system technology by handling cases where rules and procedures are invalidated by unanticipated events.

The knowledge representation architecture (figure 2) will reason about faults in the following manner. The system to be diagnosed is first modeled causally in a hierarchical sense with each level of the hierarchy

*Technical design details are not being published in this paper because of approval difficulties. Details will be included in forthcoming publications.

showing the connection graph of the component structure. Each component is then modeled as an object which contains an executable qualitative model of its physical behavior. The component object also includes a list of component inputs and outputs, input and output value limits, connectivity information, history (this would include Mean Time Between Failures (MTBF)), and heuristics (rules of thumb) about the component. With the system structure defined to a depth adequate for the system diagnosis (appropriate Line Replaceable Unit (LRU) level), the SDE will start with the root of the hierarchy and determine which subcomponents of the root component are malfunctioning. The SDE will prioritize the list of suspect components using knowledge about connectivity, heuristics, knowledge of system goals, component history and resource limitations. The SDE will then successively call for diagnosis of these components. Finally, the lowest level of the hierarchy will be reached and control would be passed to another portion of the SDE application containing system function knowledge to determine the recovery steps that are necessary.*

Recovery reasoning requires an understanding of the functions a system is required to carry out. This understanding should include the relationship of these functions to the standard goals of the system, to other functions (i.e. functional interdependencies) and to the components that make up the system. A possible abstract depiction of this representation is shown in figure 3. This representation would come into use after the diagnosis had identified the faulty component. The functions that are dependent on faulty components would need to recover either through functionally redundant hardware or a change in the system operations. The exact nature of the functional representation and the reasoning processes that utilize it is future work that must be completed as an extension to the general diagnostic methodology developed today.

4.0 PROTOTYPE DEVELOPMENT

To provide demonstration and proof of concept, the SDE was applied to a small example application; the Manned Maneuvering Unit (MMU) Translational Hand Controller (THC) was chosen for this purpose. (The MMU is the backpack used by the astronauts during Space Shuttle Extravehicular Activities). The following related tasks are currently complete: 1) a minimal core capability of the SDE implemented in KEE on a Symbolics computer, 2) a qualitative simulation of the MMU THC capable of failing one or more components and implemented in KEE, 3) a knowledge base for the MMU THC application (a graphical representation of the structural description for the MMU THC is shown in figure 4 and the structural description derived from the MMU THC knowledge base by the SDE is shown in figure 5). (11)

Major goals of the SDE implementation include maintaining portability and reasonable execution speeds. If automated diagnostics is to be transferred to real world systems, portability is an important issue. The projects mentioned in section 2.0 and the SDE itself have been prototyped using a variety of powerful software tools and computers. These environments are intended to be rapid prototyping environments and are unlikely to be ported to the final physical system for use in integrating intelligent diagnostics. The SDE has implemented methods, representation structures, algorithms, etc. with the intent to port to hardware and software that can be integrated into physical systems. Although the porting task could require substantial recoding, (e.g. porting to Ada), the underlying design will remain unchanged.

5.0 CONCLUSIONS

A requirement of automated FDIR is the integration of the system design with the intelligent FDIR software. This integration will benefit the system development during all phases of the life cycle by providing 1) the capability to analyze diagnostic instrumentation requirements during the design phase, 2) providing a ready software architecture for implementation of intelligent diagnostics, and 3) providing the opportunity to analyze different instrumentation configurations for failure coverage necessary in safety critical operations. Automating FDIR procedures for a physical system that is already designed can result in difficult problems and unsatisfactory results. Although some level of automation can be obtained for the physical system that is already in use, FDIR operations will probably always require the attention of a human operator (e.g. reading a volt meter as described in section 3.0). If intelligent automation is to be implemented successfully and fully autonomous FDIR capability is desired, it is necessary to integrate intelligent FDIR automation during all phases of the system life cycle. The System Diagnostic Engine (SDE) methodology allows an integrated approach to development of intelligent FDIR.

ACKNOWLEDGEMENT

I wish to thank my co-designer who contributed greatly to this work but has requested to remained anonymous for this publication.

REFERENCES

1. Abbott, K., Schutte, P., Palmer, M., Ricks, W., "Faultfinder: A Diagnostic Expert System with Graceful Degradation for Onboard Aircraft Applications", 14th International Symposium on Aircraft Integrated Monitoring Systems, Friedrichshafen, West Germany, September, 1987.

2. Barrow, H., "VERIFY: A Program for Proving Correctness of Digital Hardware Designs", QUALITATIVE REASONING ABOUT PHYSICAL SYSTEMS, MIT Press, Cambridge, Massachusetts, 1986, 437-493.

3. Bobrow, D., "Qualitative Reasoning About Physical Systems: An Introduction", QUALITATIVE REASONING ABOUT PHYSICAL SYSTEMS, MIT Press, Cambridge, Massachusetts, 1986.

4. Bowes, R., Cambell, T., "A Model-Based Approach To MIL-STD-1553 Verification And Diagnosis", American Helicopter Society National Specialists' Meeting on Flight Controls and Avionics, Cherry Hill, New Jersey, October, 1987.

5. Davis, R., "Diagnostic Reasoning Based On Structure And Behavior", Artificial Intelligence, Elsevier Science Publishers B. V. (North-Holland), 24, 1984, 347-410.

6. de Kleer, J., Williams, B., "Diagnosing Multiple Faults", Artificial Intelligence, Elsevier Science Publishers B. V. (North-Holland), 32, 1987, 97-130.

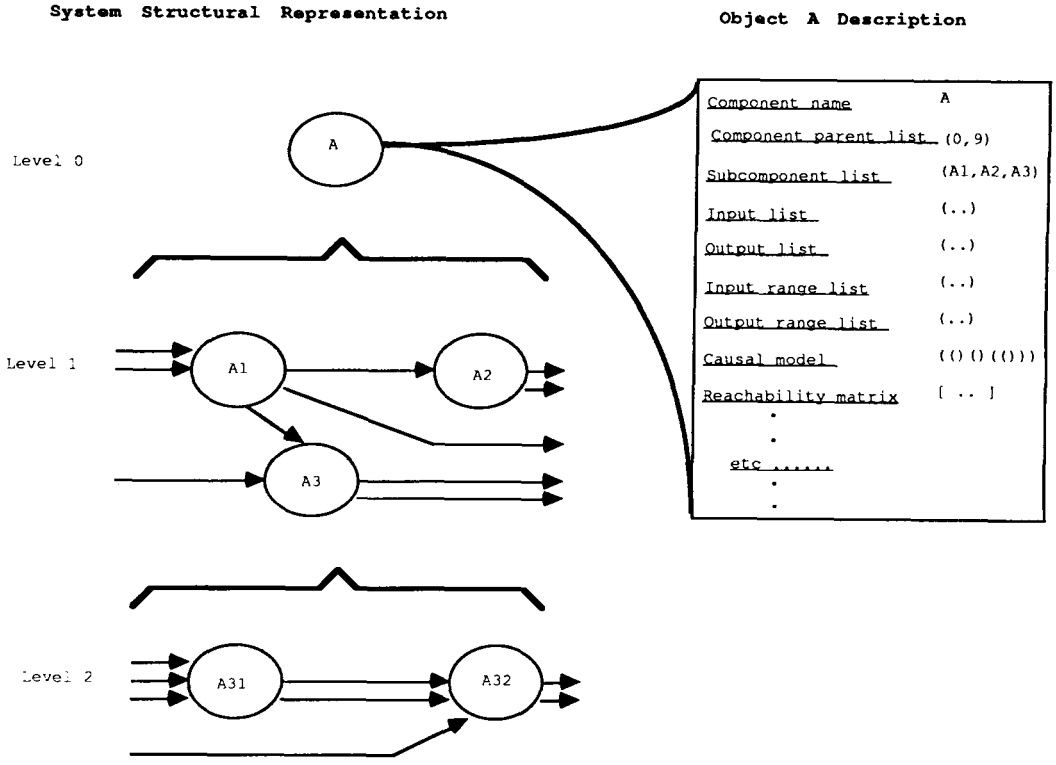
7. Hamilton, T., "HELIX: An Application of Qualitative Physics to Diagnostics in Advanced Helicopters", AAAI Workshop on Qualitative Physics, Urbana, Illinois, May, 1987.

8. Kuipers, B., "The Limits Of Qualitative Simulation", IJCAI 85, Los Angeles, California, August, 1985.

9. Reiter, R., "A Theory Of Diagnosis From First Principles", Artificial Intelligence, Elsevier Science Publishers B. V. (North-Holland), 32, 1987, 57-59.

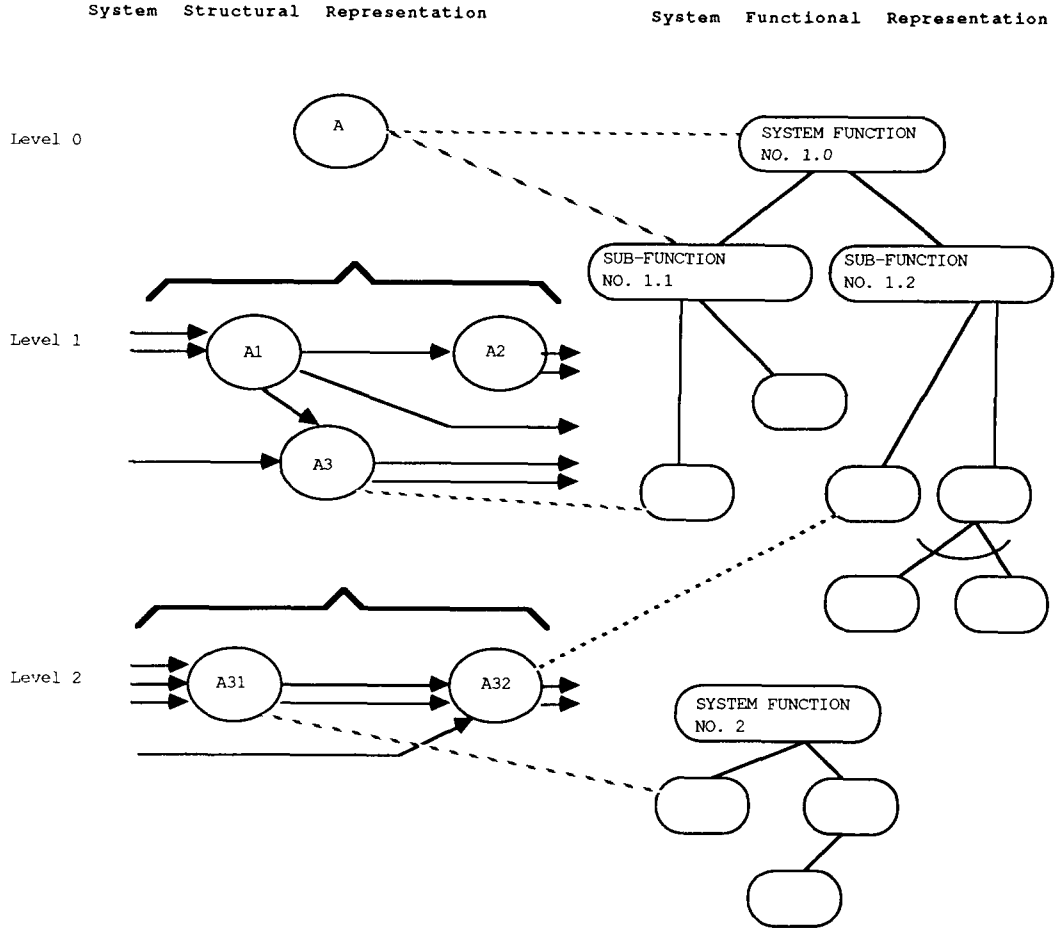
10. Williams, L., Lawler, D., "Diagnosis: Reasoning From First Principles and Experiential Knowledge", Annual Workshop on Space Operations, Automation and Robotics, NASA/JSC, Houston, TX, August, 1987.

11. Williams, L., Lawler, D., "MMU FDIR Automation Task, Final Report", Contract NAS9-17650, Task Order EC87044, Crew and Thermal Systems Division, NASA/JSC, Houston, TX, February, 1988.



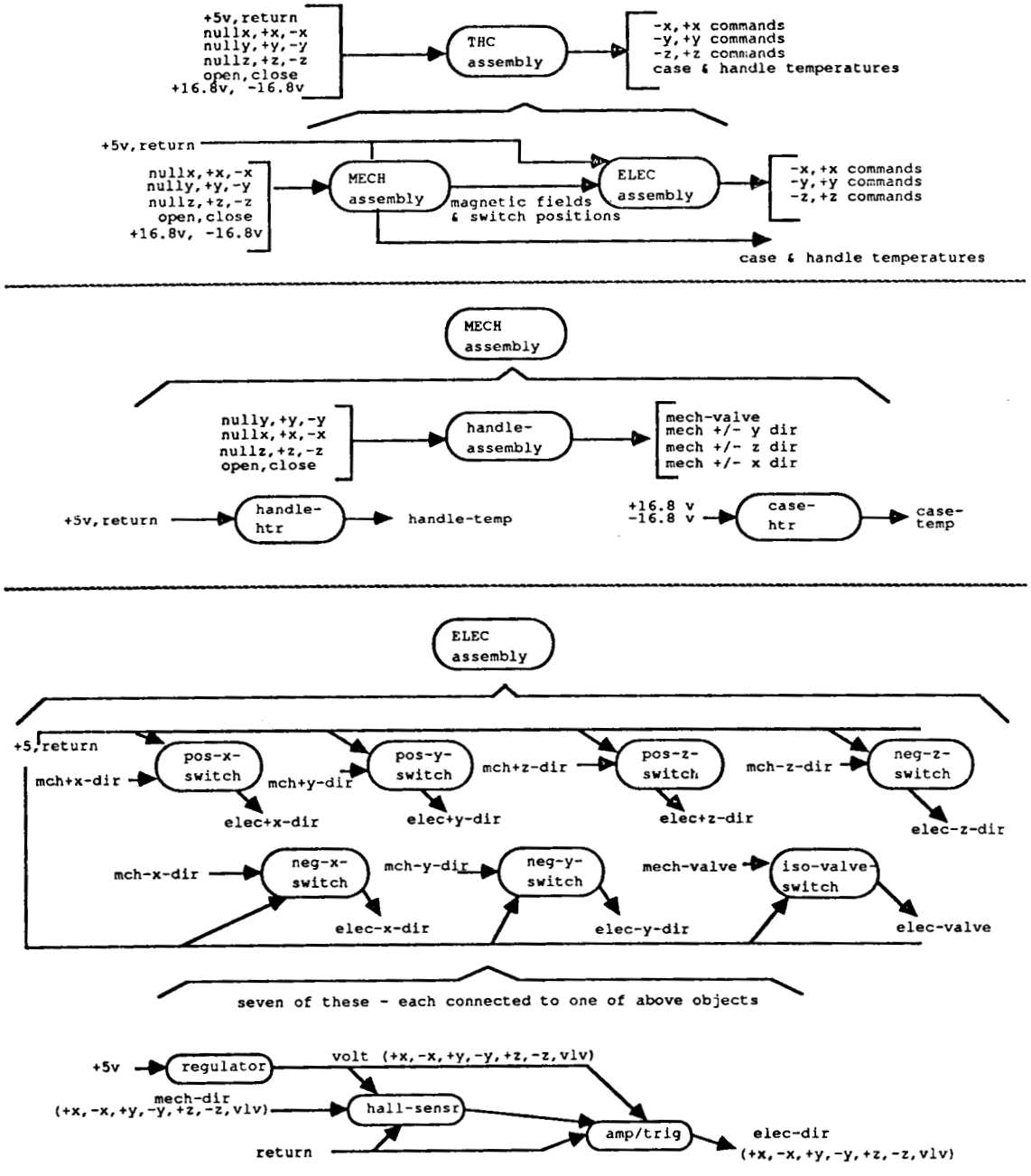
The system is modelled as a hierarchy of directed graphs to whatever depth is needed to provide enough detail for diagnosis. The nodes of the graphs represent the subsystems, assemblies, components, subcomponents, etc. of the overall system. Each node has an object representation (or frame) that contains all information pertinent to the diagnosis.

Figure 2 Knowledge Representation Architecture



Possible future extensions of the knowledge representation architecture. The system functionality is specified as an AND/OR graph (not fully-connected). Links are then established between the system structural hierarchy and the functionality graph. These links indicate which system objects are required for the system to be able to carry out particular functions or subfunctions.

Figure 3 Possible Structural/Functional Mapping



Hierarchical breakdown used to represent THC stucture. The top level breaks down into mechanical and electrical assemblies, the 2nd and 3rd level break electrical and mechanical into lowest level physical components.

Figure 4 MMU Hand Controller Structural Architecture

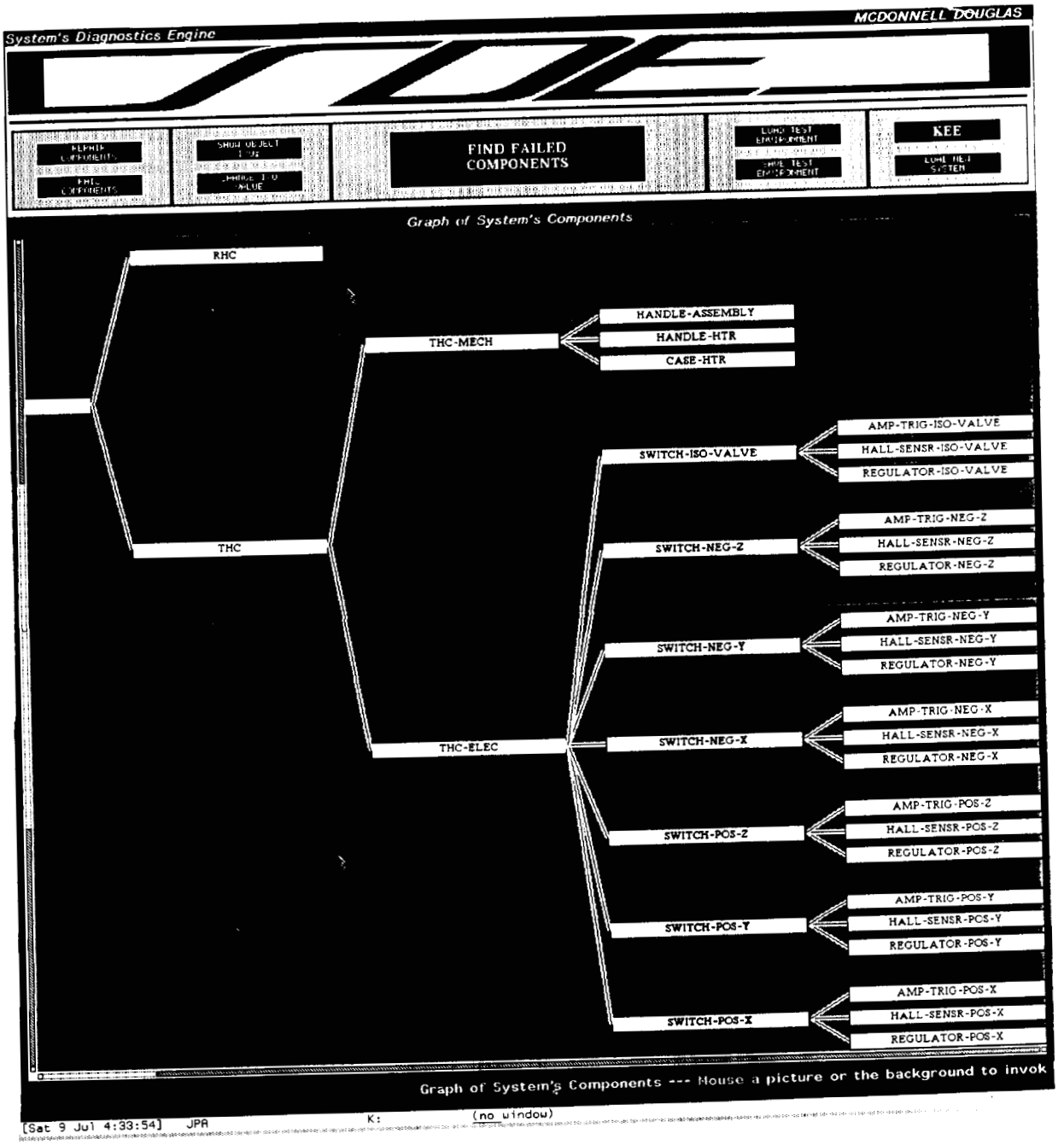


Figure 5 SDE Derived Structural Description of MMU Hand Controller