Function-based Modeling and Troubleshooting

Robert Hawkins, Jon Sticklen, & James K. McDowell
Intelligent Systems Laboratory
A714 Wells Hall
Michigan State University
East Lansing, MI, 48824
hawkinsr@cps.msu.edu
sticklen@cps.msu.edu
mcdowelj@cps.msu.edu

Tim Hill
& Roger Boyer
Thermal Advanced Automation
16055 Space Center Blvd.
McDonnell Douglas Aerospace
Houston, TX, 77062
hill@kbs.mdc.com

Abbreviated title: Function-based Modeling and Troubleshooting.

Send proofs to:

Jon Sticklen,
Computer Science Department
A714 Wells Hall
Michigan State University
East Lansing, MI, 48824

1. A version of this paper was presented at the AAAI workshop on Reasoning About Function, held in Washington, D.C. on July 11, 1993
Abstract: This report describes an ongoing effort to apply the functional modeling (FM) approach to the representation of and reasoning about engineered artifacts. The application domain is the External Active Thermal Control System (EATCS) of space station FREEDOM. The intuitions behind FM are three-fold. First, knowing the purposes of a device allows organization of the causal understanding of how a device works. Second, causality may be represented in modular chunks which are indexed by the purposes of the device and its interrelated subsystems. Finally, the global behavior of a device in a given situation can be understood by composition of the relevant causal net fragments. These starting intuitions provide a framework for organizing calculations about a device and for performing a limited type of simulation with the organized ensemble. Parallel to the FM modeling effort for EATCS is the development of a methodology that generates diagnostic knowledge for a device directly from its FM representation. This knowledge compilation strategy takes the FM representation of the target device and compiles organized knowledge structures for troubleshooting. The knowledge structures are utilized in two ways for diagnosis: 1) efficient hypothesis generation and 2) hypothesis evaluation. The central result of the research reported here is that an FM representation of a device can be used to generate knowledge for diagnostic troubleshooting."
1.0 Introduction

The functional approach to device representation (FR) originated at the Ohio State University with the 1983 research of Sembugamoorthy and Chandrasekaran (Sembugamoorthy & Chandrasekaran, 1984). FR has subsequently been extended and applied in a number of ways and across a number of domains. In the Intelligent Systems Laboratory at Michigan State University, our primary emphasis to date in this research track has been to develop simulation techniques utilizing functional device representations (Sticklen, Chandrasekaran, & Bond, 1989). The combined representational/reasoning approach we have called the “functional modeling” (FM) viewpoint to distinguish it from other FR motivated research.

The underlying intuitions of FM are three fold. First, knowing the purposes of a device allows organization of the causal understanding of how a device “works.” Second, in FM, causality is represented in modular chunks which are indexed by the device’s purpose. This is in contrast to many causal network approaches where the device is represented monolithically by a causal net at single layer. Finally, the global behavior of a device in a given situation can be understood by the composition of relevant causal net fragments.

The organizational structure provided by FM has been utilized to support diagnostic problem solving (Sticklen & Chandrasekaran, 1989). Previously, a FM model was integrated with a compiled level reasoner to determine the consequences of potential malfunctions. This knowledge, generated at run-time by consequence finding, was then used to evaluate the potential malfunction hypothesis. The compiled level reasoner directed the FM based reasoner on an as-needed basis to fill in gaps in the compiled knowledge.

The following report directly details in part a newly initiated track in which the functional viewpoint is utilized to support troubleshooting. This effort differs from both the simulation and diagnosis efforts that utilized FM in the past. This new track of research is motivated by the needs of our long term research partners at the McDonnell Douglas Aerospace Corporation. In collaboration with MDC in St. Louis, we have previously modeled
the fuel system of the F/A-18 aircraft utilizing FM (Pegah, Sticklen, & Bond, 1993). Moving from the F/A-18 fuel system, our current target application was selected in collaboration with McDonnell Douglas Space Systems, Thermal Control System Group, and is aimed at providing trouble shooting capability without the use of any experience-based knowledge.

The testbed domain of the research is the External Active Thermal Control System (EATCS) of space station FREEDOM. This track of research contains two parallel efforts: 1) modeling the EATCS using the FM representation and 2) developing a compilation methodology that produces diagnostic knowledge directly from a FM of an engineering device. This diagnostic knowledge can then be used for troubleshooting. The important result of these efforts is once the modeling of the EATCS is complete, knowledge for efficient troubleshooting can automatically be generated.

In Section 2.0, the FM approach to device representation is described briefly. In Section 3.0, the progress of the modeling effort specific to the EATCS is presented. Our new diagnostic approach within the context of FM is described in Section 4.0. Section 5.0 gives a detailed example of the diagnostic approach using an FM model of an Adder-Multiplier. Section 6.0 closes the paper with conclusions and the direction of the research.

2.0 Device Representation: the Functional Approach

2.1 Intuitions

The functional approach to device representation and reasoning begins with the intuition that if we know the purposes of a device then we can use them to organize our causal understanding of the device. This provides an organization that uses these purposes to index causal descriptions of behaviors that achieve these purposes. FM decomposes the complex causal knowledge of a device along functional lines and then for a particular set of condi-
tions composes these modular causal packages into a situation specific causal account of the device behavior.

To aid in perspective, the FM approach can be compared to related approaches to device modeling. In causal network approaches, device knowledge is represented by a network structure and inferencing involves controlled navigation of this network. In contrast, the naive physics approach centers on deriving the potential functionality (i.e. the desired behavior) of a device from the behavior and structure of its constituent devices, based on a temporal precedence view of causality. However, naive physics traditionally does not incorporate the teleological concept of function. The FM approach proposes an intermediate position between the reliance on deriving all large scale cause-effect relations of naive physics and the compiled representation of all cause-effect relations of the causal network approach. The FM approach can be thought of as a hierarchical organization of causal network fragments indexed by the device’s functions and the functions of its components.

2.2 Background

The FM approach springs from the broad framework of the Generic Task theory of knowledge based systems (Chandrasekaran, 1983; Chandrasekaran, 1986), extends the earlier framework of Sembugamoorthy and Chandrasekaran (Sembugamoorthy & Chandrasekaran, 1986), and builds on our initial conception of how a compiled level problem solver and a deep level problem solver can interact (Sticklen & Chandrasekaran, 1985). From a representational perspective, FM largely adopts the original formalism for functional representation of Sembugamoorthy and Chandrasekaran.

In addition to our extensions of the functional viewpoint, others have engaged in similarly motivated research. Goel has used this approach as a basis for design and redesign problem solving (Goel & Chandrasekaran, 1989). Punch uses FM style simulation to aide diagnostic reasoning in an integrated problem solving environment (Punch, 1989). Allemand has reported the application of this approach to understanding and debugging com-
puter programs (Allemang, 1990). Keuneke has demonstrated that the functional viewpoint can contribute to the extraction of explanations for diagnostic problem solving (Keuneke, 1992). Recently Sticklen, Kamel and Bond have demonstrated that FM can be used to organize quantitative calculations of a device (Sticklen, Kamel, & Bond, 1991). Similarly motivated research focusing on device function has also been reported by Chittaro et al. (Chittaro, Constantini, Guida, Tasso, & Toppano, 1989), Franke (Franke, 1992), Bradshaw (Bradshaw & Young, 1992), Abu-Hanna (Abu-Hanna, Benjamins, & Jansweijer, 1992), and others.

2.3 Modeling Primitives

To represent a device functionally, the device is first recursively decomposed into its constituent subdevices. In engineered artifacts, this decomposition typically parallels the major structural systems of the device.

The second step in representing a device functionally is to enumerate the functions of the device and, recursively, each of the subdevices, down to the component level. A function is composed of three elements:

- a Provided clause which states the conditions under which the function will be applicable. This amounts to a precondition for the function.

- a ToMake clause which states the result which will be achieved after the function completes. The ToMake clause may be thought of as a postcondition.

- a By clause which points to the causal description of how the function is implemented. We have so far limited our functional representations to implement functions by behaviors, as described below.

Functions provide a means of abstractly knowing what can be achieved (ToMake), what must be true for a given function to be applicable (Provided), and a pointer to a causal
description of how the function is implemented (By). Below, a fourth element for function description is described: the functional role.

To complete a functional representation for a device, the behaviors which implement functions (pointed to the “By” clauses in functions) should be described. Behaviors are directed graph structures in which the start nodes of the graph are tests of state variables of the device, and other nodes are descriptions of changes in state variables. Behaviors resemble fragments of causal nets. However, unlike causal nets, the edges of the directed graph are annotated and point to an elaboration of why each node transition takes place. These annotations are either pointers to “world knowledge” or to other parts of the functional description itself; i.e., to lower level functions or behaviors.

To summarize, there are four central facets of the FM approach to device representation. First, the functional representation is a conceptual abstraction of what a device is and how it works. The “what it is” part is represented as a collection of sub-devices related by a “ComponentOf” relation (usually shown as a component hierarchy). The “how it works” is represented as the functionality of which it is capable and the behaviors that accomplish those functions. Second, a functional description exhibits a natural modularity. A sub-device of the overall device may be replaced with another totally different sub-device which accomplishes the same functions.

Third, in understanding from the top level the device functionality, we are normally led via a chain of:

device => function => behavior

=> sub-device => function => behavior...

to lower and lower levels of sub-devices. However, this path of understanding may be terminated before the lowest levels of the device are reached. Once a level is reached at
which a particular functionality of some underlying sub-device may be “assumed true,”
then further probing along the current path is unnecessary. This ability to probe only as far
as needed follows directly from the modularity of representation adopted. Put another way,
in the functional approach to device understanding, there is an implicit natural “layering of
understanding” from the most abstract levels of device description to the most detailed.
Finally, and related to the last point, each behavior in a functional representation can be
thought of as a fragment of a complete causal net. Each of the fragments carries with it (in
its start nodes) predicates which indicate when the fragment is applicable.

The points above are not unrelated. Overall, FM manages the complexity involved in
comprehending a complex device by a divide and conquer strategy; i.e., by decomposition.
The decomposition is two fold: the device-subdevice dimension, and the device causality
dimension. In the device causality dimension, fragments of causal knowledge are “behav-
iors” which are indexed by abstractly stated functions.

3.0 Modeling the EATCS using the FM Approach

3.1 EATCS Structure

The External Active Thermal Control System (EATCS) is the temperature maintenance
system for space station FREEDOM. It is a refrigeration system whose main purpose is to
remove heat from the living areas and experiments housed in the space station and reject
the excess heat into space. The EATCS is best conceived of as a thermal bus for cooling the
space station. A simple schematic is provided in Figure 1.

The EATCS consists of three main subsystems: Heat Acquisition System, Heat Rejec-
tion System, and Pump System. The three subsystems are pictured in Figure 2 and are out-
lined and labeled in Figure 1. In Figure 1, the Evaporator bank and the pipes associated
with it are part of the Heat Acquisition Subsystem. The Pump subsystem includes the Back
Pressure Regulating Valve (BPRV), the Rotary Fluid Management Device (RFMD), the
Accumulator, and the associated piping. The BPRV is a valve that (at a high level) controls system pressure by controlling the rate of vapor flow through the condensers. The RFMD is a complex device designed to use centrifugal force in low-gravity situations to pump a two-phase fluid. The rejection system is mainly composed of radiator banks and pipes. Modeling the EATCS is complicated by the intricate nature of the Pump Subsystem and by the two-phase fluid flow and heat transfer that occurs throughout the system.

3.2 EATCS Modeling

The device decomposition mirrors the subsystem description provided above. Each of these systems have been decomposed into their constituent component parts. The modeling effort (namely identifying functions and behaviors) has focused primarily on the Heat Acquisition System. The other systems, the Pump System and the Heat Rejection System, have been decomposed but have yet to be completely modeled in terms of their functions and behaviors. The modeling of the Heat Rejection System will follow naturally from the effort devoted to the Heat Acquisition System. The Pump System represents the greatest challenge because of its sophisticated design and will be modeled last in the entire effort.

During this first phase of modeling (just completed), several structural and behavioral assumptions were made to simplify the initial effort. The major structural assumption is that there is a single evaporator loop. In the actual design of the EATCS, there are several evaporator loops in the Heat Acquisition System. This assumption has several impacts. First of all it reduces the number and kinds of components that make up the system. The model includes a single evaporator (heat exchanger), a single cavitating venturi (for flow control) and necessary simple pipes for connections. The assumption removes the need for additional complex pipes like tees and joins. From a physical behavior viewpoint, calculations of flow rates and pressures are simplified with only a single loop. The decomposition of the Acquisition System is shown in Figure 3.
Functions are assigned to each device in the Heat Acquisition Component Hierarchy in Figure 3. Items may have multiple functions that involve varyingly difficult behaviors. For simple pipes like Pipe 1, the function is to convey material from one end of the pipe to the other, without any change in temperature, quality (vapor to liquid fraction) or pressure. With respect to the changes in pressure, the heat exchanger pressure drop dominates the pressure drop in the entire loop, thus the pressure drop across a pipe is considered negligible. This assumption concerning pressure will be modified appropriately as additional loops are included. In constructing the behaviors for the Heat Acquisition System several assumptions are made. The first involves the heat load gathered by the evaporator. This load is always less than the design load and prevents any superheating of the vapor. A second and related assumption is that the temperature is constant in the loop at the saturation temperature. This implies that all heat transferred is latent heat and contributes only to changes in phase (liquid to vapor) and not to changes in temperature. Behaviors for the Heat Acquisition Systems have been constructed using these assumptions and trial simulations using the FM approach have been performed successfully. An example behavior is shown in Figure 4. This behavior shows a little bit of the complexity of the EATCS. The pressure upstream from the evaporator subsystem is calculated by a function of the pressure downstream of the evaporator system.

We are now moving to systematically relax the assumptions in the first modeling stage of the Heat Acquisition System. First, the single loop will be expanded to two evaporator loops, one near and one far. The far loop will have more lengths of pipe, and the additional loop will make the use of tees and joins necessary. Once these structural modifications are in place and the simulation trials successful, important behavioral assumptions will be relaxed. The Heat Acquisition System will be allowed to gather heat loads above the design load, thus forcing the relaxation of the constant temperature assumption as well.
Once modeling of the Heat Acquisition system is complete, model-building focus will turn to the Heat Rejection System, where the dominant physical behavior is condensation rather than evaporation, and then finally to modeling the complex PumpSystem.

4.0 Function-based Trouble Shooting: FM-Dx

In a device FM, each of the abstractly-stated functions of every system component is implemented by a behavior: a directed acyclic graph in which nodes represent state changes of the device, and links represent annotations explaining why transitions from one device state to another take place. Behaviors are thus chunks of causality that are packaged and indexed by the functions of the device. The kernel idea of our diagnostic approach, which we call FM-Dx, is to build knowledge structures that can serve a central role in diagnostic reasoning. These knowledge structures are derived by inverting the indexing that exists in an FM model.

The information processing task (IPT) of FM-Dx is depicted in Figure 5. The diagnostic input to FM-Dx is a set of sensed variables which are “Out of Bounds” (OoB variables). These OoB variables are all state variables of the device which have current values which are out of nominal operating range. How the input list of OoB variables is produced (including sensor validation issues) is not a central part of this report; it is however important to understand that the information processing task shown in Figure 5 is only a part of the IPT for the entire diagnostic situation we expect for the EATCS.

The IPT shown in Figure 5 is a standard abductive problem solving picture in which the output list is a set of items which when taken collectively will account for the input manifestations. Classically, abduction is described as follows.

The essence of abductive inference is the generation of hypotheses, which, if true, would explain some collection of observed facts. (Pople, 1983)
Although some approaches to abductive problem solving, such as set covering approaches (Reggia, 1985), attempt to develop the processing map from input to abductive output in a unitary step, many approaches (e.g., (Josephson & Josephson, 1993)) rely on a two step process.

First, a set of hypotheses are generated in which each member may account for one (or more) of the input items to be explained. Call this set $H_I$. The set $H_I$ contains hypotheses which are independent from one another. Second, $H_I$ is processed (possibly with additional inputs) to produce a set $H_D$ which contains hypotheses which taken together can explain all of the input manifestations. Some approaches consider $H_D$ to be the “best” explanation, while others consider it to be satisfying. The major point however is that in many abductive approaches, abductive IPTs such as that in Figure 5 are accomplished in the general two step process described above.

As described in Section 2.0, there are three major knowledge types within an FM device representation: a hierarchical decomposition of device components, a listing for each component of the engineering design purposes for each component (*function*), and a detailed causal description of state variable changes for how each function is achieved (*behavior*). The FM-Dx algorithms use only the third type of knowledge to compile knowledge structures to support abductive problem solving.

The central approach in FM-Dx is to begin with the represented behaviors of an FM device model, and from that input construct dictionaries which capture two relationships between device components, and device state variables:

- $R_{\text{evokes}}$ — which is a map between a device state variable and the device components which affect that variable, and
- $R_{\text{accounts-for}}$ — which is a map between a device component and the device state variables it affects.
These two relations are basically the same as those used by Pople to describe the pioneering INTERNIST medical diagnosis system (Pople, 1977). The only significant difference between Pople’s terminology for the clinical medicine situation and ours for the engineered artifact situation is that Pople uses the term “patient manifestation” while we use the term “device state variable.”

For Pople, $R_{\text{evokes}}$ and $R_{\text{accounts-for}}$ are relations which are experience-based and are thus determined by debriefing clinical practitioners. In DM-dx, $R_{\text{evokes}}$ and $R_{\text{accounts-for}}$ are computed from an FM device representation. Recall that within an FM represented device model, each behavior (as described in Section 2.0) consists of a causal sequence of state variable changes. Thus, an FM device representation directly contains a linkage between behavior and state variable changes.

The $R_{\text{accounts-for}}$ map is derived from three components. For some component $c$, with a set of functions $\{F\}$ and a set of behaviors $\{B\}$ which implement $\{F\}$, the $R_{\text{accounts-for}}$ map from $c$ to

1. Those state variables that are directly set in one of the behaviors in $\{B\}$

   This is straightforward requiring nothing more than looking up a given behavior description.

2. Those state variables that are referenced in functions/behaviors that are invoked (by one of the link annotations) in one of the behaviors in $\{B\}$, and recursively so.

   This is basically saying that a behavior $b$ in a high level component of a device should be considered as affecting the state variables of all functions/behaviors which are invoked by $b$ and recursively applied.

3. Those state variables which are computed using state variables that are set in one of the behaviors of $\{B\}$.
In the FM modeling approach, we allow one state variable to be calculated in an arbitrary way from other state variables (Sticklen et al., 1991).

Unlike the situation in clinical experience-based medicine, in our device models, $R_{\text{accounts-for}}$ is the inverse of $R_{\text{evokes}}$. Having developed one of the maps, it is easy to develop the inverse map. These two maps $R_{\text{accounts-for}}$ and $R_{\text{evokes}}$ can be however computationally expensive to produce. It is easy to see that item 2 just above would in general require computations of exponential complexity.

It is important now to keep firmly in mind what the inputs are for FM-Dx and what type of output is desired. The input to FM-Dx is a list of out-of-bounds state variables of the device. Note carefully that this assumes that some other agent is responsible for taking raw sensor data from the device, and converting into a set of state variables which are currently out of nominal ranges. Note also that the input form does not distinguish between “out of bound HIGH” or “out of bounds LOW”. We have initially avoided all but the most simple type of input to allow the compilation of $R_{\text{evokes}}$ and $R_{\text{accounts-for}}$ to be conceptually straightforward.

The diagnostic procedure of FM-Dx is shown below in Table 1. Notice that the applications of the two mapping relations (Steps 2 and 4) in the procedure of Table 1 are table lookup operations given the existence of the mapping relations $R_{\text{evokes}}$ and $R_{\text{accounts-for}}$. Step 5 is a computational cheap step as well. Step 1 is a simple input step and it too is computational simple. That leaves Step 3 as the potentially major contributor to the computational complexity of FM-Dx at run time – as distinguished from the compile time complexity of constructing $R_{\text{evokes}}$ and $R_{\text{accounts-for}}$ from an FM model.

The complexity of Step 3 depends almost exclusively on two factors, the cardinality of $\{H_1\}$ and the value of $N_{\text{max-faults}}$. Although it may not be in general true, the maximum number of expected simultaneous faults in the EATCS situation is of order magnitude 10, excluding common mode failures. Again emphasizing that this is not a general result, this
in turn means that the algorithm for FM-Dx will be efficient at diagnose time for the EATCS.

We have tested the diagnostic algorithm of Table 1 on a previously implemented testbed FM model: an idealized automotive cruise control system (Sticklen et al., 1991). The testing showed overall the diagnostic technique we have developed is promising. We have not yet tested FM-Dx against the FM model of the EATCS. We will do this when a working model of the Heat Rejection subsystem (with assumptions listed in Section 3.2 on page 9 removed) is achieved.

5.0 Diagnostic Example

5.1 Functional Model

To provide insight into how the diagnostic knowledge in an FM model is compiled into the dictionaries $R_{\text{evokes}}$ and $R_{\text{accounts-for}}$ discussed in Section 4.0 on page 11, we provide an example FM model and the diagnostic information that FM-Dx would compile from it. This example is quite simple and is shown here for pedagogical purposes. The example that we will use is an Adder-Multiplier from (Davis and Hamscher, 1988). Figure 6 shows a schematic of the device. The purpose of the device is to output two calculated values, I and J, based on five inputs, A through E. Intermediate values between the subdevices of the Adder-Multiplier are labeled F, G and H. A functional subsystem decomposition of this device is shown in Figure 7. The decomposition of the device is based on the functions of the device. The device has two main functions - calculating I and calculating J. Group1 consists of the subdevices who participate in calculating I, and group2 consists of those subdevices which help calculate J. Multiplier2 is considered separately, since it contributes to both high level functions. In the terms of the modeling primitives of Section 2.3, the function Calculate-I has:

- a $Provided$ clause of: Provided Values exist at A, B, C, and D.
• a ToMake Clause of: Calculate I as f(A, B, C, D)

• a By Clause of: By Behavior1

Behavior1 for the Adder-Multiplier is shown in Figure 8.

5.2 Diagnostic Compilation

The $R_{\text{evokes}}$ and $R_{\text{accounts-for}}$ dictionaries for this device are shown in Figure 9. Adders and Multipliers are abbreviated as Mx and Ax to simplify displaying the information. From the behavior in Figure 8, it is straightforward to notice that the variable I being in the OoB set evokes the possibility of Multiplier 2 or Group 1 being faulty, because their functions are involved in the function that calculates the value at I. This gives us devices Multiplier1, Multiplier2 and Adder1 of Figure 6. It can also be recognized that Multiplier2 accounts-for values G and I being in the OoB set, because both of those values are set causally downstream from Multiplier2 in Behavior1. Notice from the schematic in Figure 6 that Multiplier2 also could account-for value J being OoB. This makes sense since Multiplier2 is involved in the calculation of the value at J, but it is not shown in the example behavior. That knowledge would be compiled from a different behavior in the FM model of the Adder-Multiplier, under the Calculate-J function.

5.3 Diagnosis

In Figure 10, we see the application of the diagnostic algorithm to the Adder-Multiplier device. Our input set (OoB) is sensor locations G and I. The maximum number of faults ($N_{\text{max-faults}}$) is 3. We apply $R_{\text{evokes}}$ and get the set {$M1, M2, A1$}, then take the power set of this set up to $N_{\text{max-faults}}$. This set is $H_D$. Next, we apply $R_{\text{accounts-for}}$ to $H_D$ to get the list of OoB variables that each component list in that set would account for. Next, as in step 5 of Table 1, “Diagnostic Algorithm of FM-Dx,” on page 24, we evaluate the list based on explanatory power. Since the hypothesis M1 A1 accounts-for F and I alone, we would remove it from our list of potential hypotheses, since it does not account-for the OoB vari-
able G. We would, however leave the M2 A1 hypothesis, since it does account-for both the G and I OoB variables. In this example, this final hypotheses set is M2, M1 M2, M2 A1, M1 M2 A1.

6.0 Conclusions and Future Research

FM-Dx is a middle-ground approach between traditional compiled diagnosis and model-based diagnosis. FM-Dx is efficient because compiled structures are used during diagnosis as opposed to tracing and simulating the device model at run-time. The knowledge in the compiled structures is derived from the FM model of the device, so in a sense it is model-based knowledge. The driving motivation in the design of the approach is to produce efficient run-time diagnosis in the context of real-time diagnosis and repair of critical space station components.

There is continuing discussion concerning the relationship between knowledge compilation and model-based reasoning. Researchers in knowledge compilation have sought organizational principles for structuring knowledge and methodologies for gathering knowledge for various problem solving activities (diagnosis, design, planning). Model-based reasoning has focused on the direct manipulation of representations of structure and behavior in the pursuit of problem solving. Some researchers in knowledge compilation view the models used in MBR as sources of knowledge, much like domain expert protocols, that then can be converted into knowledge structures for problem solving. This notion has met some resistance in the MBR community and the utility of this notion has been debated in the literature without an apparent community-wide consensus (Davis, 1989; Keller, 1990; Keller, 1991). Keller (Keller, 1991) identifies two types of transformations that may be performed on models to generate problem solving knowledge. The classification is based on how the knowledge in the model is modified. Content-preserving transformations attempt to re-organize knowledge contained in a model into a more suitable form for problem solving. This can be done by several methods; Keller suggests simplification,
macro-operators, precomputation and partial evaluation as possible content-preserving transformations. Content-modifying transformations change the content of knowledge in the model by adding knowledge or by removing knowledge. Transformations in this class could include threshold application, quantitative to qualitative transformation, information elimination and functional approximation.

In its current form, FM-Dx is a content-preserving transformation for knowledge compilation from a model. The problem solving knowledge contained in the two mapping relations is derived directly from the model without additional knowledge. More precisely, FM-Dx is a type of precomputation. However in this case the result is not a numeric one, but a knowledge structure representing the results of tracing through the behaviors of the FM model. The savings from the FM-Dx compilation comes in that the model is NOT traversed during run-time problem solving to generate and evaluate hypotheses. The intermediate inferences involved in such traversals are avoided as now hypothesis generation and evaluation knowledge is directly accessible from the derived knowledge structures. The important facet of content-preserving transformation is that in as far as the FM model is correct, the FM-Dx knowledge structures will be correct. This is not the case in content-modifying transformations, where the assumption of the transformations can affect the knowledge structures correctness.

Keller demonstrates several of the notions concerning the compilation of models in an example focused on the reaction wheel assembly (RWA) of NASA’s Hubble Space Telescope (Keller, 1991). In the example Keller uses both content-preserving and content-modifying transformations to generate fault localization rules from a structural and behavioral model of the RWA. The target knowledge structures of the transformation (fault localization rules) can be directly applied and involve very little problem solving inference (direct match).
FM-Dx differs from Keller’s compilation example in that FM-Dx uses only content-preserving transformations. FM-Dx also differs from Keller’s in the form of the target knowledge structures. The target knowledge structures for FM-Dx are designed to support abductive problem solving, which as described earlier is a two stage inference in our approach.

There are a number of issues that we intend to focus on now to further develop FM-Dx. One of the most important is to redo our approach to take into account “out of bound HIGH” and “out of bounds LOW” situations. We have avoided any elaboration of state variable values beyond simple “out of bounds” to make the diagnostic algorithm particularly simple, and the compilation necessary to support it conceptually straightforward. The result is a working diagnostic approach, but one that casts “too wide a net.” Our intuition is that by distinguishing HIGH and LOW variable situations, that we will obtain a more specific diagnostic result at least in some situations.

Secondly, a natural complement to the diagnostic algorithm is to provide an explanation of the diagnostic results. Because of the derived nature of the compiled knowledge used in diagnosis, an inference path back to the FM model is available. In this way the FM model of the device can be used in conjunction with the diagnostic algorithm to generate an explanation of the problem solving inferences.

FM-Dx continues the development of knowledge compilation of model representations and provides an example of how effective and efficient diagnostic knowledge can be generated from a model. The debate within the MBR community mentioned earlier can only be settled with continued exploration, development and analysis. FM-Dx continues this path and provides some initial results within the viewpoint that compilation of knowledge structures from models is a useful exercise.
7.0 Acknowledgments

Dr. Bill Bond, McDonnell Aerospace Corporation, St. Louis, helped a great deal in initial understanding of the EATCS. Dr. Ron Rosenberg, Mechanical Engineering Department, Michigan State University, spend many hours as well in helping us to understand the EATCS.

Research reported here has been supported in part by funding from the McDonnell Douglas Research Laboratories’ Independent Research and Development funds. The broad research to extend and apply functional reasoning techniques in the Intelligent Systems Laboratory is also supported by ARPA (ARPA 8673) and the NSF Center for High Speed Low Cost Polymer Composite Processing at MSU. Generous equipment support for the MSU Intelligent Systems Laboratory has been provided by Apple Computer. The FM modeling software of the Intelligent Systems Laboratory is implemented in Parc Place Systems Smalltalk, version 4.1, VisualWorks.

8.0 References


Table 1. Diagnostic Algorithm of FM-Dx

1. Input the set of out-of-bounds variables \( \{OoB\} \) and the maximum number of simultaneous, independent faults which will be considered \( N_{\text{max-faults}} \).

2. Apply \( R_{\text{evokes}} \) to each element of \( \{OoB\} \) and merge the resulting sets to produce a set of independent hypotheses, \( \{H_I\} \), containing candidate components that may be malfunctioning.

3. Form a set of possible abductive answers \( \{H_D\} \) which is composed of all elements of the power set of \( \{H_I\} \) excluding those elements of cardinality greater than \( N_{\text{max-faults}} \).

4. Each potential abductive answer, that is each element in \( \{H_D\} \), consists of a set of named components of the device. Apply \( R_{\text{accounts-for}} \) to each component to obtain the OoB variables that component would account for, and continue to form the composite list of OoB variables that each abductive answer will explain.

5. Rank the abductive answers on explanatory power (how many OoB variables each explains) and select the highest ranked as the most plausible set of broken components in the device.
Figure 1: Schematic of EATCS
Figure 2: EATCS System Decomposition

Excess Heat from Living Areas and Experiments

EATCS

Heat Acquisition System

Pump System

Heat Rejection System

Deep Space

heated fluid

cooled fluid
Figure 3: Subsystem decomposition of Heat Acquisition System
Figure 4: Example Behavior for Heat Acquisition Subsystem

- $Pressure-A > 0$
- $Current-Load > 0$

By Function of Pipe 1 - Transfer Pressure

Set $Pressure - B$ to $Pressure - A$

By Function of Evaporator Subsystem - Change Quality

Set Quality $- D$ to $F(Load, flow-rate, ...)$

By Function of Pipe 2 - Transfer Quality

Set Quality $- E$ to Quality $- D$

By Function of Pipe 2 - Transfer Pressure

Set Flow-Rate $- D$ to Flow-Rate $- C$

By Function of Pipe 2 - Transfer Flow

Set Flow-Rate $- E$ to Flow-Rate $- D$

By Function of Evaporator Subsystem - Transfer Pressure

Set $Pressure - D$ to $Pressure - E$

Set $Pressure - C$ to $F(PressureD, Load...)$
Figure 5: Information processing task of FM-Dx
Figure 6: Schematic of Adder-Multiplier
Figure 7: Functional Device Decomposition of Adder-Multiplier
Figure 8: Behavior 1 for Adder-Multiplier

Values exist at A and C

Set G to \( f(B, D) \)

By Function of Group 1
ADD/MULT

Values exist at B and D

By Function of Multiplier 2
Multiply

Set I to \( f(G, A, C) \)
Figure 9: $R_{\text{evokes}}$ and $R_{\text{accounts-for}}$ Dictionaries for Adder-Multiplier

$R_{\text{evokes}}$

- I >> A1, M1, M2
- J >> M2, M3, A2
- F >> M1
- G >> M2
- H >> M3

$R_{\text{accounts-for}}$

- M1 >> I, F
- M2 >> I, J, G
- M3 >> J, H
- A1 >> I
- A2 >> J
Figure 10: Diagnostic algorithm applied to Adder-Multiplier

\[ R_{\text{evokes}} \]

- [G],[I] \rightarrow M1,M2,A1
- \downarrow
- M1,M2,A1, M1 M2, M1 A1, M2 A1, M1 M2 A1

\[ R_{\text{accounts-for}} \]

- M1 >> F, I
- M2 >> G, I, J
- A1 >> I
- M1 M2 >> F, G, I, J
- M1 A1 >> F, I
- M2 A1 >> G, I, J
- M1 M2 A1 >> F, G, I, J

- \downarrow
- M2, M1 M2, M2 A1, M1 M2 A1