NRC Publications Archive
Archives des publications du CNRC

A Perspective on Explanation in Diagnosis

Abu-Hakima, Suhayya

This publication could be one of several versions: author’s original, accepted manuscript or the publisher’s version. / La version de cette publication peut être l’une des suivantes : la version prépublication de l’auteur, la version acceptée du manuscrit ou la version de l’éditeur.

Publisher’s version / Version de l'éditeur:

Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Explanation and Problem Solving, 1993

NRC Publications Archive Record / Notice des Archives des publications du CNRC :
https://nrc-publications.canada.ca/eng/view/object/?id=adb278da-8c06-46dc-8c89-377df61870da
https://publications-cnrc.canada.ca/fra/voir/objet/?id=adb278da-8c06-46dc-8c89-377df61870da

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L’accès à ce site Web et l’utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D’UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n’arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research Council Canada
Conseil national de recherches Canada

A Perspective on Explanation in Diagnosis

Suhayya Abu-Hakima
Knowledge Systems Laboratory
Institute for Information Technology
Building M-50, Montreal Road
National Research Council of Canada
Ottawa, Canada
K1A 0R6
email: suhayya@ai.iit.nrc.ca
tel: (613) 993-8551
fax: (613) 952-7151

Problem: ***Diagnosis***
Fault-Based Diagnosis *versus* Model-Based Diagnosis
Method: Generate and Test *versus* Cover and Differentiate
Model: Functional *versus* Structural

Summary

This position paper argues that causal explanation in diagnostic tasks are more easily achieved in fault-based or failure-driven reasoning versus model-based reasoning. Fault-based or failure-driven diagnosis is more of a contextual task and can more easily be used to support user interaction through explanation than model-based diagnosis. A problem with fault-based reasoning that is often cited is the difficulty in reasoning about novel faults. In response to this key problem researchers in diagnostic systems are examining model-based reasoning. The latter approach is computationally expensive and in an effort to make it more efficient it is sometimes combined with fault-based reasoning techniques. The paper elaborates on explanations in RATIONALE and reasoning in JETA, both fault-based diagnostic systems. It also describes model-based diagnosis. Some of the issues and approaches needed to achieve causal explanation in model-based reasoning by integrating it with fault-based reasoning conclude the paper.

Fault-Based Diagnosis

Fault-based reasoning (FBR) is used in many diagnostic systems. Knowledge in FBR is largely based on maintenance manuals and interactions with experts intended to capture heuristic knowledge about the maintenance and repair of a device or process. The knowledge in these systems is often represented as rules or frames which are organized into troubleshooting hierarchies such as that shown in figure 1. At the top level of the hierarchy is the general knowledge representing a problem with the device. This general problem is refined systematically until the leaf nodes of the hierarchy which represent physical repairs or adjustments to the device are reached. Once these repairs are achieved by a human technician some diagnostic systems re-test to confirm that the symptoms and diagnosed faults are cleared through backtracking in the hierarchy.

FBR systems have evolved considerably since the development of MYCIN [Shortliffe et al. 77]. MYCIN was developed to provide advice treatment for microbial infections. MYCIN inspired several follow-on systems such as NEOMYCIN [Clancey 86] which had an expanded knowledge base and EMYCIN, a shell based on MYCIN. Clancey also used

EMYCIN to develop GUIDON, an intelligent tutoring program for diagnosis [Clancey 86]. The MYCIN programs started with rules and evolved into meta-rules in NEOMYCIN which provided some structure to an otherwise flat knowledge base. These programs also advocated the use of certainty factors which are probabilities attached to the rules and combined to provide a measure of certainty or uncertainty to a diagnosis. The MYCIN approach remains a very widely used approach in FBR systems as exemplified by the literature review of [Abu-Hakima 93].

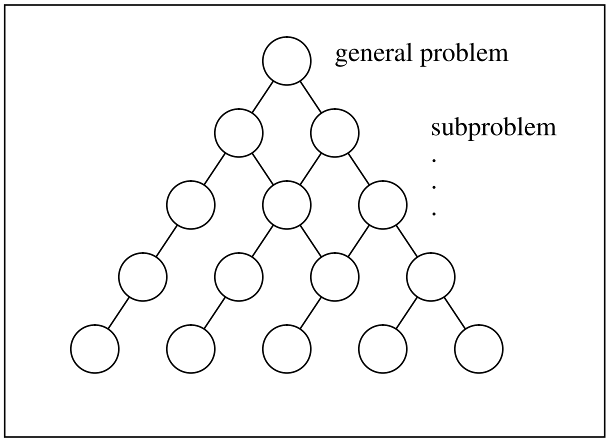

Figure 1. Structured diagnostic hierarchy

Classificatory problem solving is contextual in nature. Diagnosis is often referred to as classification. Chandrasekaran and his colleagues developed MDX, a system that diagnosis a form of liver disease, cholestasis [Chanrasekaran et al. 79]. MDX has a diagnostic hierarchy which is referred to as a conceptual hierarchy since it guides the reasoner in the global sense through diagnoses clustered as concepts that establish local contexts. Local uncertainties and knowledge are used to guide the diagnosis [Chandrasekaran and Tanner 86]. MDX has served as a model for many well-structured diagnostic systems including RATIONALE and JETA. RATIONALE is a workstation diagnosis system that establishes context in reasoning so that it may support the user with sophisticated explanations of diagnoses that help justify system behaviour and clarify reasoning [Abu-Hakima 88]. In RATIONALE, many of the ideas advocated by Chandrasekaran for structuring FBR systems and handling uncertainty are applied. This approach was found to be ideal for explicitly representing knowledge so that it may be explained [Abu-Hakima and Oppacher 90]. RATIONALE diagnosis faults with Xerox workstations. It generates dynamic and static template-based explanations that include why, how and what-if responses. Explanation remains a major objective of FBR systems and most systems have why and how explanation but do not necessarily generate hypothetical (what-if) ones.

RATIONALE

In RATIONALE domain knowledge is hierarchically organized into problems and subproblems. Reasoning proceeds by constructing a hypothesis-based classification tree whose root hypothesis contains the most general diagnosis of the system. Guided by a focusing algorithm, the diagnostic hierarchy (classification tree) branches into more specific hypotheses that explain the more detailed symptoms provided by the user. As the system is used, the hierarchy forms the basis for a dynamically generated explanation hierarchy which holds both successful and failed branches of the reasoning tree.

RATIONALE is designed to reason explicitly for the purpose of clearly explaining its reasoning strategies, i.e. its inference engine coincides with its explanation module. Explicit reasoning is supported by knowledge as to: why a particular hypothesis is preferred in a given situation to an alternative; why exceptions may be overruled in some situations; and why global inference strategies succeed in certain circumstances but fail in others. This explicit reasoning is what easily provides RATIONALE with its capability to establish local and global contexts which are used for explanation. By aligning the functioning of the inference engine with that of the explanation module, the explainer has immediate access to all the domain knowledge and strategic information that drives the reasoner. To align the functioning of the inference engine with that of the explanation module, the hierarchy of domain hypotheses in RATIONALE is represented by frames, as illustrated in Figure 2 (note that part of the hierarchy shows a subset of the workstation domain as an example). At the root of the hierarchy is the top level goal or the overall aim of a particular knowledge-based system domain. RATIONALE could be used to define a multi-domain diagnostic system with each domain having a unique aim and a unique hypothesis tree. The top level goal serves as an initial entry point to the domain under consideration. For example, if several domains were defined in RATIONALE, e.g. a heart, a liver and a kidney domain, and a patient's symptoms related to a heart problem, then only the domain hierarchy for the heart problems would be searched for a hypothesis explanation.

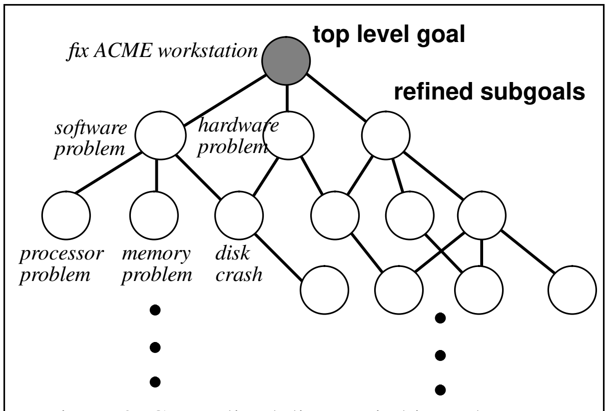

Figure 2: Generalized diagnostic hierarchy

The subgoal frame (see [Abu-Hakima and Oppacher 90] for a frame template) includes three explanation slots for the hypothesis it represents. The novice, experienced and expert explanation slots are input by the knowledge engineer when the subgoals are being defined. An explanation detail slot particular to the hypothesis, holds the expert's numeric estimate of the conceptual complexity of the hypothesis and of its explanatory importance relative to the other hypotheses in the domain hierarchy. This provides a customizable range of detail to distinguish between user levels to output explanations. As RATIONALE reasons explanation slots are used to output contextual advice for determined diagnoses.

The set of all possible symptoms for a hypothesis are grouped to form an enabling, an invalidating and a tolerance subset. These subsets are matched against the user's observed symptoms and their conditions are applied according to the degree of match between the expected symptoms and the ones observed. Each subgoal has an alternate-subgoal slot which is activated when a set of symptoms is shared with another hypothesis. A refinement slot points to the subgoal to be refined next and to the sufficient symptoms required for its refinement. The enabling, invalidating, alternate and refinement slots provide RATIONALE with the capability to reason in a causal manner. This causality provides the flexibility to explain to the user within a local or global context why diagnoses are established, invalidated or are refined. It also allows the user to examine alternate diagnoses so that hypothetical explanations may be generated for a currently established symptoms.

RATIONALE was designed to provide four types of explanation. The first two types are session sensitive, and they are event and hypothetical explanations. The second two types explain system capabilities, and they are ability and factual questions. Note that event questions, unlike system capability questions, are dynamic and thus are displayed only after the user has started a session so that a context has been established. Some of the questions that RATIONALE supports are shown in Figure3 (see [Abu-Hakima and Oppacher 90] for the detailed set).

Generated explanations in current knowledge-based systems remain clumsy and hide the reasoning strategies. Illustrations of RATIONALE’s ability to handle contextually-dependent references are included below to demonstrate the strength of its template-based explanations. Templates connect pieces of text to variables that are instantiated from the knowledge in the system. This allows explanation templates to be domain independent. Templates also simplify the task of generating dynamic explanations according to the current context.

Event questions:
For my current problem, why do you need these symptoms?
For my current problem, why do you NOT need these symptoms?
For my current problem, why did you conclude this problem?
For my current problem, why did you NOT conclude this problem?

Hypothetical questions:
What-if these symptoms were also true for my current problem?
What-if this problem replaced my current problem with these symptoms?

Capability questions:
How do you deduce this problem?
How do you refine this problem?

Figure 3: Some RATIONALE question types

Contextually Dependent References

Once a context is introduced in a user question, the generated explanation should reference it much as a human explaining within a particular context would. (Note that '/' separates the template phrase choices - the choices are made based on the current context of the session.)

User Question: Why did you not deduce *subproblem B*?
RATIONALE Explanation template:
Subproblem B could not be deduced since /
the alternate subproblem symptoms S_a, of current *subproblem A* could not be concluded/
the sufficient focusing symptoms S_s of current *subproblem A* could not be concluded /
none of the selected symptoms could lead to it.

Workstation Example
User Question 1: Why did you not deduce a *hardware problem*?
RATIONALE: A *hardware problem* could not be deduced since the alternate subproblem symptom *memory error*, of the current *software problem* could not be concluded.

User Question 2: Why did you not deduce a *processor software problem*?
RATIONALE: A *processor software problem* could not be deduced since the sufficient focusing symptoms *failed boot* and *memory error* of the current *software problem* could not be concluded.

The tools used to implement RATIONALE are Quintus Prolog and HyperNeWS 1.2 [van Hoff 91]. Prolog is used to implement the reasoning and explanation modules while HyperNeWS is used to implement the user interface.

JETA

The Jet Engine Troubleshooting Assistant (JETA) is a tool developed to assist a technician in diagnosing aircraft engines using a hypermedia interface which provides contextual help. For a diagnostic application to properly support hypermedia, one requires a structured manner by which to represent the knowledge, reason about it interactively, display it dynamically and explain it to the user (see [Abu-Hakima et al. 93] for a thorough description of JETA's hypermedia interface). JETA's knowledge representation and reasoning strategies are more flexible than those of other diagnostic systems including RATIONALE's [Halasz et al. 92].

JETA's troubleshooting knowledge is represented as a diagnostic network that is hierarchical in nature. Each node in the network corresponds to a decision point in the troubleshooting process and the links represent relations directing the flow of control between nodes. The overall network is much broader than it is deep since there are many components and associated symptoms. The number of nodes along a network path varies from four to twelve in a network of approximately 200 nodes. Possible next moves in the network are represented as children of a node. Any node can have multiple parents since a component malfunction may be due to many causes. The troubleshooting knowledge is encoded at each diagnostic node using a custom command language. In JETA as in RATIONALE, advice generating slots are included in the frame and their contents are output to the user as diagnoses or procedures to follow to find a fault. In JETA, advice is supported with a schematic or a graph. An indexed database of schematics and graphs is kept so that only pointers to the database are kept in the frame. The current implementation of JETA links text, graphs and schematics in this manner.

JETA supports two other types of frames, diagnostic parameter frames and glossary frames (see [Halasz et al. 92] for JETA's frame templates). Diagnostic network frames have activation rules that fire based on the values of diagnostic parameters. Glossary frames hold the definitions of terms output to the user in hypermedia interactions involving the diagnostic network frames or the parameter frames.

The reasoning strategy, while interpreting JETA's frame command language, addresses those strategies inherent in engine diagnosis. It blocks entry to a particular node if certain pre-conditions have not been cleared. Logical jumps in the network are permitted in order to abort the current line of reasoning should new observations lead to other nodes becoming more probable. Nodes are activated by rules that depend on network status and relevant parameters. Postponement of certain procedures at various nodes is allowed by a technician who may have a ‘hunch' that a certain path is incorrect, or who may be limited by time or a parts shortage. Dynamic contextual help, graphics and hypertext that describe nodes or terminology is provided. Advice is customized at the node level before it is displayed to the user. Allowances for instrument mis-readings or failures are made. Nodes can be re-visited if it seems that a faulty component has been used. Parameters are updated as the diagnosis extends over time. For example, temperature will change during the day making initial values invalid for later tests requiring the same information. Looping is controlled in the network by setting and monitoring node states. The reasoner lets the user select from a list of prioritized tests and failure modes. The list order is based on the difficulty of a test and the likelihood of the failure and can be dynamically re-ordered depending upon observed symptom combinations. The diagnostic nodes in the network assume states that are monitored by the reasoner. This controls the search by pruning the network and provides trace information to track sessions for explanation generation.

The user interface module accepts input from the user and passes it to the reasoning module for processing. It also receives output from the reasoning module and displays it for the user. The interface is designed for three types of users: the maintenance technician, the system manager and developers of JETA. This distinction is necessary since technicians require an interface tailored to their maintenance tasks whereas developers require special access to the knowledge-base and reasoning module. A system manager has access and authority to update lists of user names and passwords as well as engine serial numbers.

JETA's tiled user interface has a reasoning process that intervenes only when the user requests assistance (see [Abu-Hakima et al. 93] for example screens). It provides glossary and context-sensitive help throughout a diagnostic session that a novice technician may use continually whereas a seasoned technician may access infrequently. User input is made by menu and iconic button selection with a mouse. The glossary and definition functions which are cross-referenced meet the objective of providing context-sensitive information and automated cross-referencing. The interface uses hypermedia links between graphics and text to provide the user with hypermedia-based advice.

The four tools used in JETA’s development are: Common LISP, PostScript, NeWS and HyperNeWS 1.4. [van Hoff 91]. The reasoning strategies and knowledge-base representation are implemented in LISP. The multiple stack interface runs under HyperNeWS, which itself runs under the Sun Micro Systems Network Windowing System (NeWS). NeWS runs under the Unix operating system.

Model-Based Diagnosis

Model-based reasoning (MBR) for diagnosis concentrates on reasoning about the expected and correct functioning of a device (figure 4). A device is modelled based on its components and their expected behaviour [Hamscher and Struss 90]. Such models range from quantitative ones to qualitative ones and all attempt to approximate device behaviour as accurately as possible. Once a device model is stabilized then a device's observed behaviour can be predicted from the model. If a discrepancy in behaviour is detected then possible candidates based on assumed component faults are generated. These candidates are generated based on assumptions that describe correct model behaviour. Sequential diagnosis is used to choose observations, augment a prediction for the candidate faults and update the list of candidates until a dominant candidate is found.

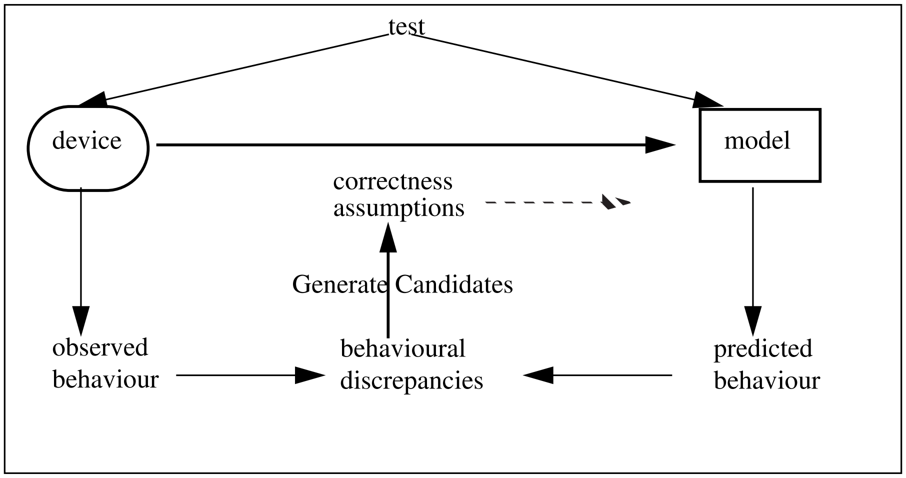

Figure 4: Model-Based Reasoning for Diagnosis

In MBR there are many conflicting definitions for models. They range from causal models represented as semantic networks with links specifying the relations between component nodes to full blown numerical simulations for complex systems and processes that have taken decades to perfect. Generating models is a key problem in MBR. Some researchers generate causal models, others generate models with structure and behaviour while others generate functional models for devices.

Davis was one of the earlier proponents of MBR. In [Davis 84] he describes a theory to exploit reasoning on the basis of device structure and behaviour. He defines paths of causal interpretation. He also describes constraint suspension used to identify which components are responsible for which faults. He argues that we need to balance complexity versus model completeness in diagnosis thus we need to enumerate and layer categories of failure. He also advocates the use of the relaxation of assumptions to bridge faults in the topology itself. This is one of the landmark papers in the area of model-based reasoning. Quite a bit of work has followed Davis' examples and theories.

De Kleer and Williams published a landmark paper on MBR for diagnosis describing GDE, the General Diagnostic Engine [de Kleer and Williams 87]. GDE infers behaviour from device structure and functionality. It is applied to digital circuits and makes use of an ATMS (Assumption-Based Truth Maintenance System). This work forms the cornerstone of ATMS-based model-based reasoning systems. It was followed by many papers that criticized the approach as not computationally practical in diagnosing faults with large complex systems. Some of the papers criticizing GDE propose the use of hierarchical fault-based reasoning to reduce the computational complexity of de Kleer and Williams' approach. Struss in [Struss 89] lists four shortcomings of de Kleer's GDE:

- an inability to deal with a large number of components,
- no temporal reasoning capabilities,
- missing fault knowledge and
- uncertain measurement handling.

He then describes how GDE+ handles: simple dynamic aspects, multiple tests, hierarchical knowledge and unreliable observations. He enhances GDE to include hierarchical device subsystem reasoning, temporal reasoning, the unknown fault assumption to address intermittent faults and uncertain measurements. GDE+ is a migration back to heuristic or empirical diagnoses using fault-based reasoning. Struss points out that neither GDE nor GDE+ address: changing device structures, complex temporal behaviour (feedback), uncertainty or the use of qualitative models in reasoning. In [Struss and Dressler 89] the authors describe GDE+'s ability to exploit contradictions between the assumed correct behaviour of components and observations. They also analyze whether faultiness of components would explain observations. The paper advocates the representation of a fault view for each component. It points out that a fault and a healthy view (state) for a component cannot be true in the same time instant (consistent belief rule). It also gives the 'no good inference rule' where the node and its opposite which represents a fault cannot be true at the same instant. The ATMS is then modified to reason with the fault as well as the no-fault behaviour of a device. Their work gives excellent insight into combining model and fault-based diagnosis to deal with GDE's shortcomings.

Other MBR authors have argued about the definition of device functionality versus behaviour. Sticklen in [Sticklen et al. 88] describes modelling a device's functionality by:

- decomposing the device into sub-devices,
- stating abstractly the functions, goals and purpose of the device and
- representing the manner of achieving the device functions, goals and purpose.

This functional modelling can be likened to distributing the causal knowledge of the device. In naive Physics changes in the small are used to derive changes in the large. In a causal network changes are woven into a comprehensive net. This provides a detailed device representation strategy for device components and functions and their inter-relations. Instead of trying to derive device and component behaviour (using naive physics) one tries to use device. In [Malin and Liefker 91] a good definition of functionality is given which argues that function is the set of goals the device is intended/designed to achieve. Thus to model it one needs to model the structure, behaviour, goals and timing of the device. Structure is device components and their interrelations. Device behaviour is what the device does. The approach has been applied to the space shuttle remote manipulator system as a demonstration for how it is applicable to complex problems.

Causal Explanations: Integrating Fault and Model-Based Diagnosis

This position paper has described two approaches to fault-based or failure-driven diagnosis. RATIONALE's approach focuses on making knowledge representation and reasoning explicit so that system behaviour may be justified to the user. JETA focuses on supporting the user through a hypermedia interface with a somewhat less explicit approach to knowledge representation and reasoning. Both approaches have merits and drawbacks. RATIONALE's reasoner is not as flexible as JETA's. JETA cannot explain its behaviour as explicitly as RATIONALE since causality is more difficult to establish in JETA. However, both approaches support the user with contextual help while a diagnosis is achieved.

It is more difficult to visualize contextual help in model-based diagnosis. If we follow the de Kleer [de Kleer and Williams 87] approach which represents a device with functionality as a set of components with behaviour. The device can be diagnosed by assuming a faulty component and enumerating the behavioural states that the fault propagates in the remainder of the device. This is compared to the behaviour that a technician is observing in attempting to isolate the problem. Model-based diagnosis can detect novel faults since the behaviour of the device is the basis of its knowledge representation and reasoning. Fault-based reasoning uses the faults of a device rather than its actual behaviour, hence it cannot detect novel faults. However, model-based reasoning can lead to a combinatorial explosion in producing a diagnosis for complex systems (for example, an aircraft engine) and it does not lend itself to explanation.

Given the simplicity of generating explanations for fault-based diagnostic systems as exemplified by RATIONALE and the flexibility in reasoning about faults while providing contextual help in JETA, a natural progression would be to combine model and fault-based diagnosis in order to benefit from both approaches. This is an idea that is actively being pursued in diagnostic research. GDE+ models both the fault and correct behaviour of the components of a device [Struss and Dressler 89]. If the GDE+ approach is extended further to include an explicit hierarchy of the faults of a device as in RATIONALE and JETA, one achieves an important level of integration between MBR and FBR. It is essential to maintain a single diagnostic engine for both MBR and FBR. This implies that the same domain knowledge is used for reasoning and explanation. A single reasoner would manipulate the domain knowledge with two views or control strategies: one that reasons with faults and another that reasons with the intended or predicted behaviour of the components of the device. Ideally common faults would be reasoned with using the FBR view and novel faults would require the MBR view. The FBR view would be explained in the manner already described for RATIONALE and JETA. The MBR view would require the expansion of explanation to include more of a functional explanation strategy. The explanation would proceed by relating the expected functions of the components to the observed symptoms which in turn imply component failures that had been diagnosed.

My current research is in the automatic generation of a functional model of a device from its fault-based knowledge. By extracting a functional model both fault and model-based diagnosis can be pursued in a single system gaining from the advantages of the two approaches while minimizing the disadvantages. My application is in the area of complex electromechanical devices, specifically jet engines.

References

[Abu-Hakima 88] Abu-Hakima, S., *RATIONALE: A Tool for Developing Knowledge-Based Systems that Explain by Reasoning Explicitly.* Masters thesis, Carleton University, Ottawa, Canada. May 1988.

[Abu-Hakima and Oppacher 90] Abu-Hakima, S., and Oppacher, F., Improving explanations in knowledge-based systems: RATIONALE. Knowledge Acquisition journal (2). 301-343. December 1990.

[Abu-Hakima 93] Abu-Hakima, S., Diagnostic techniques in knowledge-based systems: a review of approaches, applications and issues. 100 p. To be published as an NRC ERA (1993).

[Abu-Hakima et al. 93] Abu-Hakima, S., Halasz, M., and Phan, S., An Approach to Hypermedia in Diagnostic Systems. To appear as Chapter 10 in Intelligent Multimedia Interfaces, Edited by Mark Maybury, AAAI Press (1993).

[Chandrasekaran et al. 79] Chandrasekaran, B., Gomez, F., Mittal, S. and Smith, J.W., “An approach to medical diagnosis based on conceptual structures”. Proceedings of the Joint Conference on Artificial Intelligence, 134-142.(1979).

[Chandrasekaran and Tanner 86] Chandrasekaran, B. and Tanner, M.C. “Uncertainty handling in expert systems: uniform vs. task-specific formalisms”. In Kanal L.N. and Lemmer J., editors, Uncertainty in Artificial Intelligence, North Holland Publishing.(1986).

[Clancey 86] Clancey, W.J., “From GUIDON to NEOMYCIN and HERACLES in twenty short lessons: ORN final report 1979-1985”. The AI Magazine, pp. 40-60,(1986).

[Davis 84] Davis, R., “Diagnostic reasoning based on structure and behaviour”, Artificial Intelligence, Vol. 24, pp. 347-410 (1984).

[de Kleer and Williams 87] de Kleer, J., Williams, B.C., “Diagnosing Multiple Faults”, *Artificial Intelligence*, vol. 32, (1987).

[Halasz et al. 92] Halasz, M., Davidson, P., Abu-Hakima, S., and Phan, S., JETA: A Knowledge-based Approach to Aircraft Gas Turbine Engine Maintenance. Journal of Applied Intelligence, 2, pp. 25-46, Kluwer Academic Publishers, NRC 31832, (1992).

[Hamscher and Struss 90] Hamscher, W. and Struss, P., “Model-Based Diagnosis”, *AAAI-90 Tutorial Notes*, Received at the eighth national conference of artificial intelligence, Boston, Massachusetts. July 29, 1990. pp. 1-179, (1990).

[Malin and Liefker 91] Malin, J.T., and Leifker, D.B., “Representing functions/procedures and processes/structures for analysis of effects of failures on functions and operations”, Proceedings of 1991 NASA Goddard Conference on Space Applications of Artificial Intelligence, Greenbelt, Maryland, pp. 141-151, (May 13-15 1991).

[Scott et al. 77] Scott, A.C., Clancey, W.J., Davis, R. and Shortliffe, E.H., “Explanation capabilities of production-based consultation systems”. Stanford University, Dept. of Computer Science, Report No. STAN-CS-77-593,(1977).

[Sticklen et al. 88] Sticklen, J., Chandrasekaran, B., and Bond, W.E., “Distributed causal reasoning for knowledge acquisition: a functional approach to device understanding”, 3rd AAAI sponsored Knowledge Acquisition Workshop for Knowledge-Based Systems, Banff, Canada, pp. 34-1 to 34-18, (1988).

[Struss 89] Struss, P., “New Techniques in Model-based Diagnosis”, *Proceedings of Knowledge-based Computer Systems*, Bombay, India, (11-13 December 1989).

[Struss and Dressler 89] Struss, P., Dressler, O., “Physical Negation - Integrating Fault Models into the General Diagnostic Engine”, *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, Detroit, MI, (20-25 August 1989).

[vanHoff91] van Hoff, A.A., HyperNeWS 1.4 Manual. Turing Institute. Glasgow, Scotland. 1991.

Author's work on knowledge-based systems:

I have been a researcher in the area of explanation in knowledge-based systems since 1985. I have published my Master's thesis on RATIONALE - a tool for developing knowledge-based systems that explain by reasoning explicitly - which I discussed at the explanation workshop at AAAI '88. Some of the ideas for its hypermedia interface were discussed at the IJCAI '89 workshop on intelligent interfaces. My Ph.D. work is on the use of fault-based reasoning and explanation traces to model the functional behaviour of a system. I lead the development of JETA, the Jet Engine Troubleshooting Assistant, which has a hypermedia interface and sophisticated knowledge browser. I am currently researching integrated approaches to diagnosis and the development of knowledge-based tools that are application-task independent. I have recently joined the editorial board of the International Journal of Man Machine Studies (IJMMS).