A Hierarchical Symptom Classification for Model Based Causal Reasoning

C. N. Lee, P. Liu, S. J. Clark, and M. Y. Chiu

Siemens Research and Technology Laboratories
105 College Rd. East, Princeton N.J. 08540

ABSTRACT

Model based causal reasoning has been widely used for physical systems diagnosis. The system fault is localized with the causal relation of system structure and behavior. In such applications, if the system fault is not localized with the observed behavior, then a subsequent observation is made. This research studies *a hierarchical symptom classification* for guiding a subsequent observation in model based causal reasoning. The diagnostic symptoms are mapped to the system functional hierarchy and the symptoms are classified by partitioning the functional hierarchy. The dependency relation of symptoms guides subsequent observation. This strategy enhances the control of subsequent observation by hierarchically structuring and classifying the symptoms.

1. Introduction

Diagnosis is an incremental information processing for localizing system faults through cycles of observing and analyzing system behavior[1 2 3] Model based causal reasoning has been widely used for physical systems diagnosis to localize system faults with the causal relation of system structure and behavior[4 5 6] In such applications, the system fault is generally recognized by monitoring the discrepancy between observed behavior and expected behavior. The behavioral discrepancy is explained by causalities of component failure. However, since system behavior is observed incrementally as diagnosis proceeds, the *symptom analysis* and *subsequent observation* are necessary in each reasoning cycle i.e., observed symptoms are analyzed and if the cause of symptoms is not explained clearly, then a subsequent observation is made†

† Fault recognition and adequacy of diagnosis detail as initial and final steps of diagnosis are also important but are not discussed here.

In our previous research[7 8] a method of dynamic causal model construction was developed for incremental symptom analysis. The method relaxes the limitation of static causal model by providing various causal models of different abstraction levels. However, if the observed symptoms are insufficient to localize a fault, then additional observation† or system measurement is needed. This research studies a symptom classification strategy for guiding subsequent observation. This strategy is based on the system functional hierarchy i.e., physical systems are designed hierarchically by functional level and then realized into physical structures. The symptoms can be mapped to the related functions and classified using the dependency relation of system hierarchy. The dependency relation of symptom classes guides a subsequent observation.

This strategy is formalized for model based causal reasoning by combining two reasoning approaches: diagnosis from first principles and diagnosis from heuristics[1] The heuristics map symptoms to the system functional hierarchy. Then the first principles described in the functional hierarchy of the system guides symptom classification to a subsequent observation. This study presents a procedure for combining two reasoning approaches and making them complement each other. This strategy is generalized as a domain independent control procedure i.e., when no other heuristics are available in diagnostic process, symptom classification based on functional hierarchy guides subsequent observation.

In the next section, natures of physical systems such as functional level design, physical realization, functional dependency, and system behavior are conceptually defined to formalize the diagnostic problem. In section three, symptom to functional hierarchy mapping is discussed and the symptom classification strategy is explained with an example. Related works and conclusions are discussed in section four and five respectively.

† We use the word observation to include the meaning of word measurement.

2. Nature of Physical Systems

Diagnostic reasoning for physical systems is forming a causal network to explain system behavior using its structural and functional description. Though normal system behavior may be derived from functional and structural descriptions, abnormal behavior is difficult to derive. It is generally because the normal behavior of each component is designed and known, but not the faulty behavior‡. However, for diagnostic purposes it is necessary to understand how the faulty behaviors are related with physical structures and functional hierarchy. Nature of physical systems are conceptually defined in diagnostic point of view. The diagnostic boundary of causal reasoning and how the diagnostic symptoms should be used within the causal model are discussed.

2.1 Functional hierarchy and dependency

Functional hierarchy and dependency of a physical system is the key for model based causal reasoning. Functional hierarchy is formed in the system design. The systems design consists of two processes *functional design* and *physical realization.* Systems are defined during functional design and are mapped to physical structures in physical realization. Functions are designed top down. Once a function is specified in a level, it is divided into several subfunctions of next lower level. This subfunctioning process continues until each function is simple enough to be realized as a physical structure. Thus, a tree structure of functional hierarchy is formed through the subfunctioning process in functional design.

Functionality of a function is defined by the connections to other functions rather than the input output relation of the function itself. The dependency of a function is given by its functionality in the system hierarchy. If the functional hierarchy represents functional relation between different level functions, then the functional dependency represents functional relations between the same level functions. We define two basic dependency relations considering function connections *(simply) dependent* and *mutually dependent.*

> **Definition 2.1** Relations between two functions, *(simply) dependent, mutually dependent,* are
> 1) $f_1 => f_2$: if f_1 output is connected to the f_2 input and there is no other connection between f_1 and f_2, then f_2 is *dependent* upon f_1,
> 2) $f_1 <=> f_2$: if f_1 output is connected to the f_2 input and f_2 output is connected to the f_1 input, then f_1 and f_2 are *mutually dependent.*

The dependent relation is a general physical connection which is used for unidirectional data flow. The causality propagates in one direction. It is a transitive relation. As a special case, if a function has unity feedback, then it becomes a self recursive function. The mutually dependent relation is a connection for bidirectional data flow. It represents mutually recursive functions. It is a transitive and symmetric relation even though the relation does not exactly match the physical connections. If reflexivity of a function is defined without physical input output connection, then the mutual dependent relation $<=>$ is an equivalence relation. The causality in this relation is found by testing functions separately. The default relation in the same level functions is generally the dependent relation in physical systems.

Though different qualitative relations can be defined using more specific functional characteristics of a system, only the above two relations are defined as domain independent relations in our research. It makes reasoning control more independent from domain knowledge. Initially, causal reasoning is guided by the domain independent relations. Domain dependent relations are used if the detail of the local function is needed.

2.2 Functional design and physical realization

The relation between functional design and physical realization provides the behavioral explanation of the physical system. The functions are realized into a physical system and the system shows the integrated functionalities of functions as system behavior. We define the physical realization process as a mapping M from functional design F_D to physical realization P_R, i.e. $M(F_D) = P_R$.

> **Definition 2.2** A *physical realization* P_R satisfies a functional design F_D with
> 1) a *functional design* F_D is a set of *designed functions,* $F_D = \{f_1, f_2, .., f_n\}$,
> 2) a *physical realization* P_R is a set of *existing functions,* $P_R = \{f_1', f_2', .., f_p'\}$, which can directly be mapped on a physical structure,
> 3) for any function $f_i \in F_D$, there exists a set f_i'' such that $f_i'' \subseteq P_R$.

There are two important things in this relation. First, it is a surjective mapping and second, there are many various physical realizations P_Rs for one functional design F_D. P_R may be any an improper superset of F_D.

Both of the *surjectiveness* and *variety* of mapping are introduced when the functional design is materialized with physical components. The surjectiveness is caused since there is no exact one to one mapping from the abstract functional design to physical realization. Extra functions always coexists in a physical realization. This is true even for a simple physical component. For example, suppose that power on off function is needed. Then all the functions of a power switch, such as knob, holder, and enclosure are added to the system even though they are not necessarily designed. The variety of mappings is caused by many different components which have the same required functions and also have some different extra functions. The different extra functions of components cause different mapping ranges and make the physical realizations different. For example, different type power

‡ Deriving abnormal behavior from functional and structural descriptions has limitations because of physical structure realization process. However, Faulty behavior is regarded as normal behavior in fault tolerant system or in built in test design.

switches can do the same on off operation.

This aspect of physical realization makes the representation problem interesting. With the same function, different physical structures may be represented. The different structure may be diagnosed with the same function. Physical realization is an irreversible process in the sense that the initial functional design can not be recovered from the physical realization. Thus, both of the functional hierarchy and its physical realization should be clearly defined if any relation between functional hierarchy and physical realization is to be used for abstracting functionality from physical realization.

2.3 System behavior and diagnosis

The system behavior is defined by the physical system structures. If some structure is changed in the system, then the system behavior may be different. Thus, diagnosis of physical systems is to find the changed part of the structure using the original system structure, behavior and the changed behavior. We conceptually define *system behavior, physical systems diagnosis* and *symptom* as follows with the definition of structure and behavior in [4].

> **Definition 2.3** *Normal system behavior* is the behavior of a physical realization P_R with existing functions of $\{f_1', f_2', .., f_p'\}$, and *abnormal system behavior* is the behavior of a physical realization P_R which has one or more abnormal function $\neg f_i$ in existing functions of $\{f_1', f_2', .., f_p'\}$.

> **Definition 2.4** Physical systems *diagnosis* is to find the difference between structure A and the changed structure A' by observing the difference between behaviors B of A and B' of A', where A, B and the relation between A and B are known, and B' is incrementally observable.

> *Symptom* S is the observed part of behavior B', which satisfies the relations of $S \cup B \neq \emptyset$, $S \not\subseteq B$, and $S \not\supseteq B$.

Suppose that a car structure A has a set of behavior B, and car structure A' has a set of behavior B'. Here, A, B and the relation between A and B are assumed to be known. If A' represent a changed structure of car A, then the difference between A and A' should be found and explained by observing the difference between B and B'. For complete diagnosis, all the behavior B' should be observed. However, since B' is only incrementally observable, it is practically difficult. If the observed behavior i.e., symptom S is insufficient then different structures may be claimed for the observed behaviors. In other words, from definition 2.3, since A represents a normal state and $\neg A$ represents a set of all possible abnormal states, A' is one of $\neg A$ and $S \subset B'$. Thus, the value of A' is not generally derivable from A and S. Also, since the relation between A' and B' is not usually the same with the relation between A and B, A' is not derivable from B'.

Furthermore, since the behavior B' and symptom S are only incrementally observable, the completeness of behavior description B' is unknown and thus the structure A' is unknown. For example, if $S \subseteq B$ or $S \cup B = \emptyset$ i.e., only normal behavior is observed as symptom or no default relation is assumed between S and B for A' then diagnosis is not available unless some more symptoms are obtained. In the case of $S \cup B \neq \emptyset$, diagnosis can not be proved to be complete because only $S \subset B'$ is true and $S = B'$ is unknown. It is undecidable whether the observed symptoms are complete or not for fault diagnosis. On this discussion we conclude the following diagnostic principle:

> **Principle 2.5** *Physical systems diagnosis problem* is underconstrained because of incremental observability and incompleteness of observation.

Therefore, only hypotheses about system state can be made to explain observed behaviors, or observations need to be made to justify the hypotheses.

We noted that no diagnosis is guaranteed from functional hierarchy and symptoms. However, physical systems diagnosis has pragmatic solutions since the very detail of diagnosis is not always required in practical cases. A circuit board in an electronic equipment can be replaced in a board level diagnosis without knowing what is exactly wrong in the board. The required detail of diagnosis decides if the solution is acceptable. Thus, by properly matching the required detail of diagnosis to the system functional hierarchy, we can get a feasible solution . In the next section, a symptom classification strategy is discussed to match the symptoms to system hierarchy for the proper level of diagnosis.

3. Hierarchical Symptom Classification

Diagnostic symptoms of physical systems are mapped onto certain failures of functions in system functional hierarchy. Since symptoms are generally related to several functions in the same level or in the different levels of functional hierarchy, hierarchical analysis of symptoms is necessary to generate reasonable hypothesis and guide subsequent observations.

3.1 Symptoms in physical systems diagnosis

Diagnosis of a physical system can be phrased as follows. The physical system is represented as a functional hierarchy and a causal model is constructed considering initial symptoms and system structure i.e., teleological functional abstraction [6,7]. The symptoms are then matched to the functions in the causal model. The observed symptoms are explained by hypothesizing the functional failures of matched functions. If the symptoms are explained with the functional failures, then the functions in the causal model are refined with the next lower level functions. The refined functions are tested with more detail hypothesis and the observed symptoms. The ambiguity in refined functions are resolved by additional observations. If additional observations or symptoms are not explained within the model, then the model needs to be modified for further diagnosis.

There are some important characteristics of symptoms, especially for physical systems diagnosis. First, since the diagnostic symptoms are mapped onto a system hierarchy, there exist algebraic set relations among symptoms. In other words, a physical system is a hierarchically designed artifact and the symptoms are generally interpreted in terms of functions in the system functional hierarchy. Thus, the symptoms have corresponding functional abstraction levels and relations. Second, since the symptoms are behavioral observations of causal effects of component failure, typically a symptom makes all the simply dependent and mutually dependent functions as suspects; this brings ambiguity[9] into diagnosis. For example, suppose a relation f1 => f2 => f3 and symptom of ¬f3. The observed symptom ¬f3 can imply ¬f1 or ¬f2. In other words, any component in the causal path can be a suspect for the symptom. Third, a component failure is propagated and the symptoms can be observed in many different places. The causal effect of a component failure in f1, which possibly is observed as ¬f1, can also be observed as ¬f2 or ¬f3. This failure propagation inherently makes observation incomplete and in turn requires additional observations. Thus, the diagnosis problem is how to minimize ambiguity and incompleteness with minimal symptom observations.

3.2 Symptom mapping and analysis

As previously discussed, physical systems diagnosis is to map symptoms to functional hierarchy and to analyze the mapping to guide subsequent observations. Though an observed symptom can be specific enough to pinpoint a component failure in the end of diagnosis process, generally it is not specific enough to locate a component failure directly during the diagnostic process. It generally can designate function blocks or a set of suspect components. It is a one to many mapping. Thus hierarchical analysis of symptom mapping is required to guide subsequent observations.

Figure 1 shows an example of symptom mapping to a functional hierarchy. Suppose that symptoms are mapped, S_1 to f21, S_2 to f441 and f442, and S_3 to f23 and f332. We used list representation for symptom mapping. S_1(f21), S_2(f441 f442) and S_3(f23(f331 f332)) represent the above examples respectively. Notice that a symptom has a range which defines the boundary of suspect functions in the hierarchy. Symptom has upper and lower limits of functional levels and the limited number of suspect functions; they are limited by the dependency of matched functions. In Figure 1, if dependent relations are assumed in each functional level except the mutually dependent relation between f331 and f332, then one can have a relation of $S_1 \Rightarrow (S_2 \Leftrightarrow S_3)$. Then the cause of the symptom S_1 should be investigated first with the relation between f21 and f22 (the strategy used is explained later in this section). To simplify our discussion, we exclude testability and probability problems in measurement[3]. These problems can be amended with measurement ordering procedure after obtaining a set of measurements.

3.3 A classification strategy for guiding measurement

The strategy of a hierarchical symptom classification is summarized as follows with the previous example:

1) Map observed symptoms to all possible functions in the system hierarchy,

2) Get symptom to function mapping lists in hierarchical order, regard mutually dependent functions as one function : S_1(f21), S_2(f441 f442) and S_3(f23(f331 f332)),

3) Do symptom dependency calculations with functional level and dependency; for level calculation,

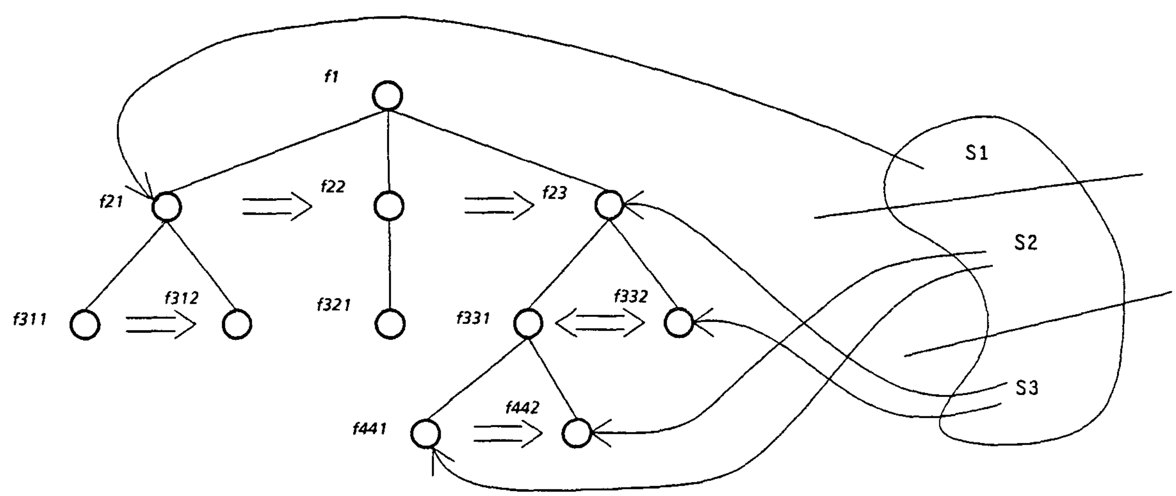

Figure 1 Symptom to functional hierarchy mapping and its partitioning

use the highest level of function if they are in one branch or mark each parent function until a dependency relation is found in the same level, and use the next higher parent's level, for dependency calculations propagate functional dependency through simply dependent and mutually dependent functions : S_2 uses f331 and S_3 uses f23, thus we know $S_2 => S_3$,

4) Construct a symptom dependency list, if possible, or get split lists : $(S_1 => (S_2 <=> S_3))$,

5) Check dependency relations and make a related measurement list, if possible : measurements about dependency relation between f21 and f23, between f441 and f442, and mutually dependent relation between f331 and f332,

6) Select a measurement from the list of step 5 and go to step 1 with the measurement result : f21 is selected since it has the highest dependency level; check if f21 is ok; if yes, check symptom S_1 or model failure; if no, measure whether f311 or f312 is malfunctioning.

There are several things to be clarified in this strategy. There is no guarantee about the completeness of symptom mapping in step 1. It is inherently incomplete because of the knowledge acquisition process from the human expert and the level of human expertise. The problem of acquisition process can be alleviated by utilizing the system hierarchy information. The mapping process can be validated and complemented by checking functional dependency and by collecting related symptoms. In step 3, if a symptom is mapped into functions with different parents, generally a measurement is needed to identify the suspect function. This introduces a measurement ordering problem. Measurement ordering is delayed until all the observed symptoms are ordered. However, if the mapped functions are independent, it may be a multiple faults case and thus the classification strategy needs to be adjusted to include multiple fault diagnosis. The multiple faults in our system are diagnosed in the operation order of system function blocks. It is reasonable assumption for a sequentially operating system.

In step 4, the symptom list is constructed based on the relation calculations of step 3. In step 5, the measurement list is constructed and the priority is given based on the symptom list of step 4. The possible measurements are assigned to each function when the functional hierarchy is constructed. If a proper measurement is not found, then the dependent relation is regarded as a mutual relation to avoid impractical measurement generation. A measurement ordering is obtained based on the symptom list. If measurement heuristics are available, then the heuristics ae applied to the list to adjust the measurement priority. In this step, the partitioning strategy can be adapted to a specific problem domain. We simply set up a procedure which utilize the size and depth of a system function block for dependency calculation of symptom group when no heuristics are available. The measurements which are related with one function are grouped together and get the same priority. In step 6, there is an interesting problem of how we propagate the measurement result. Unfortunately we do not have a general algorithm with which we can estimate maximum entropy or probabilities[10 11] of each measurement or its result in advance. For most of the measurements in the system, the resulting propagation needs too many assumptions and estimations which eventually make measurement selection impractical.

We discussed the measurement generation and ordering based on the symptoms and the system functional hierarchy. The measurements are selected and grouped based on the symptom classification. This strategy is only helpful when the system functional hierarchy is appropriately represented.

3.4. An Example

In our laboratory, a robot cell for assembly of laundry pump motors is available along with the cell functional hierarchy and its structure. The details of robot cell reasoning strategy are reported in [7] [8]. The problem is briefly described, and how our hierarchical symptom classification strategy is used for robot cell diagnosis is explained.

The symptoms are first collected through cell monitoring system and the causal model is constructed based on the collected data analysis. Based on the analysis, certain subassembly processes are suspected and a causal model is constructed considering the relations of processes. During the reasoning process, the causal model is examined to see if the observed symptom can localize fault within the model. If the symptoms are not clearly explained, then a subsequent observation is made for model refinement or for resolving ambiguity of suspects. The hierarchical symptom classification strategy is used to guide a subsequent observation. Symptom dependency is calculated considering the assembly operation steps. The robot cell hierarchy is partioned in the process level first and then assembly step level with the ordered symptoms. The observation point is selected by checking the relation between partitions. The relations between partitions are represented with the software flags or physical communication lines. Generally, the physical communication lines were selected as a suspect and tested with a sensor activation procedure. The modification and reordering of the symptom list is guided by the observation result. Since the measurement result has only the value of normal or abnormal, the result propagation was relatively simple.

4. Related Works

Causality calculation has been a main issue in diagnostic reasoning. Causality is generally calculated using a structure model[4 10] or a functionally abstracted qualitative model[6 11]. Causality in a structure model is propagated through connected components and each component has an explicit function representation. The symptoms are matched to the output of a certain structure. Thus, the integrated functionality of structure is not easily recognized and thus difficult to be matched. Causality in qualitative models propagates through qualitative variables which are carefully abstracted to represent the causal effects of integrated functions. The symptoms are matched to the states of qualitative variables which have predefined interpretation of system operations. The system functional hierarchy is explicitly or implicitly represented in the model.

In diagnostic causal reasoning [1 2 4 6 10 11], symptoms are used as stepping stones together with the system functional hierarchy. However, the symptoms in causal reasoning has been used without hierarchical notion. The symptoms are assumed to be specific enough to localize faults in a reasonable size of the system structure [4 10] or assumed to be general enough to have the same functional abstraction level for set operation [1 2]. Also, once the suspect functions or components are selected, the relations between suspects and symptoms are no longer used. Our research goal is to utilize various symptoms in different hierarchical levels. A hierarchical symptom classification strategy is developed for selecting a subsequent measurement in the physical systems diagnosis. The symptoms are classified depending upon the system functional hierarchy and proper partition of symptom classes guided subsequent observation.

The symptom classification strategy has a similar philosophy with a blackboard architecture[12] which enforces hierarchical structure for problem representation and handles constraints hierarchically. However, the blackboard architecture does not support model generation. Dynamic causal model construction strategy[7] is used together to provide a flexible hierarchical causal model generation.

5. Conclusions

The limit of physical systems diagnosis is discussed with the conceptual relation between functional design and physical realization. The relation between system behavior and diagnostic symptoms is discussed based on functional hierarchy and dependency. A hierarchical symptom classification strategy is developed for subsequent measurement selection. Symptoms mapped into the functions in system functional hierarchy are classified and ordered. A symptom partitioning, dependent upon functional hierarchy and dependency, provided a method for symptom ordering and subsequent measurement selection in the physical systems diagnosis.

The case of an implicitly modeled fault, such as bridge fault problem[4], is not discussed. It is a type of failure, which is caused by the functions which do not explicitly appear in functional design. Even though the bridge fault problem can be viewed as a case of multiple single faults in our approach, we think that it should be regarded as a problem of fault modeling rather than a problem of a symptom and structure relation. The diagnosis of this type of failure requires an extended functional hierarchy representation to describe the physical structure detail. The structure detail does not normally appear in functional design. It can be regarded as extra functionalities in physical realizations. The failure caused by extra functionality is not properly addressed in the present research.

For the future research, the relation between symptom description and system description has to be formalized. However, we are not sure whether it should be a language form or function relations in complex objects. The measurement result propagation needs to be generalized to dynamically adjust the priority of dependency relation in a symptom list. It can be achieved by appropriately specifying the problem domain. Parallel reasoning mechanism should be studied for priority calculations. The steps of symptom classification needs to be incorporated in question generation and explanation modules. Dynamic functional hierarchy generation is our long term goal.

Acknowledgements

The authors would like to thank all members of the RTL Robotics Group who shared their opinions about robot cell structure and diagnosis, special thanks to Narendra Gupta, member of the Learning and Expert Systems Group, who motivated this research and provided valuable critique.

References

[1] R. Reiter, "A Theory of Diagnosis from First Principles," Artificial Intelligence 32, 57-95, 1987

[2] J. A. Reggia, D. S. Nau and P. Y. Wang, "A Formal Model of Diagnostic Inference. I. Problem Formulation and Decomposition," Information Sciences 37, 227-256, 1985

[3] W. Hamscher and R. Davis, "Issues in Model Based Troubleshooting," M.I.T. A.I. Memo 893, 1987

[4] R. Davis, "Diagnostic Reasoning Based on Structure and Behavior," Artificial Intelligence 24, 347-410, 1984

[5] M. R. Genesereth, "The Use of Design Descriptions in Automated Diagnosis," Artificial Intelligence 24, 411-436, 1984

[6] J. de Kleer, "How Circuits Work," Artificial Intelligence 24, 205-280, 1984

[7] C. N. Lee, P. Liu and M. Y. Chiu, "Dynamic Causal Model Construction for Model Based Reasoning," Proceedings IEEE Expert Systems Conference WESTEX-87, 90-95, 1987

[8] C. N. Lee, M. Y. Chiu, P. Liu and S. J. Clark, "Model Based Hierarchical Diagnosis of Robotic Assembly Cell," Proceedings SPIE 6th Intelligent Robots and Computer Vision, Vol. 848, 182-189, 1987

[9] A. V. Aho and J. D. Ullman, "Automatic Construction of Efficient Parsers," in Principles of Compiler Design, Addison Wesley Publishing Co. Reading, Massachusetts, 198-245, 1979

[10] J. de Kleer and B. C. Williams, "Diagnosing Multiple Faults," Artificial Intelligence 32, 97-129, 1987

[11] J. de Kleer and J. S. Brown, "Theory of Causal Ordering," Artificial Intelligence 29, 33-61, 1986

[12] B. Hayes-Roth, "A Blackboard Architecture for Control," Artificial Intelligence 26, 251-321, 1985