MAD: A Real World Application of Qualitative Model-Based Decision Tree Generation for Diagnosis

Heiko Milde, Lothar Hotz, Jörg Kahl, Bernd Neumann, and Stephanie Wessel

Laboratory for Artificial Intelligence, University of Hamburg
Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
milde@kogs.informatik.uni-hamburg.de

Abstract. Computer diagnosis systems grounded on hand-crafted decision trees are wide-spread in industrial practice. Since the complexity of technical system increases and innovation cycles are shortened, the need for systematic decision tree generation and maintenance arises. In this paper, the MAD system is introduced which generates decision trees based on qualitative device models. Existing resources such as design data and expert design know-how as well as decision trees and diagnosis knowledge can easily be reused and integrated into decision tree generation. Since decision tree generation is based on device models, applying MAD reduces average fault identification cost and facilitates quality management of diagnosis equipment. Furthermore, cost of diagnosis system generation, modification and maintenance is reduced. We have successfully evaluated the MAD system in cooperation with the german forklift manufacturer STILL GmbH Hamburg.

1 Introduction

More than 100.000 forklifts made by the german company STILL GmbH Hamburg are in daily use all over Europe. In order to reduce forklift downtimes, approximately 1100 STILL service workshop trucks utilize decision-tree-based computer diagnosis systems for off-line diagnosis. Due to the complexity of the electrical circuits employed in forklifts, decision trees may consist of more than 5000 objects. When forklift model ranges are modified or new model ranges are released, decision trees are manually generated or adapted by service engineers who apply detailed expert knowledge concerning faults and their effects. Obviously, this practice is costly and quality management is difficult. Furthermore, average cost of decision-tree-based fault identification may be unnecessarily high because decision trees are not optimized. Hence, there is a need for computer methods to support systematic generation, modifications and optimization of diagnosis systems. The introduction of new diagnosis techniques, however, raises challenges.

- First, it is essential to integrate innovative with established concepts. A total redesign of existing diagnosis systems is usually unacceptable for economical reasons. In particular, for STILL, abandoning decision trees was not acceptable.
- Second, it is essential to utilize available resources such as expert knowledge and computer-based product data for diagnosis system generation. This way, the cost of diagnosis systems can be reduced and the trustworthiness of diagnosis data can be

improved.
Model-based decision tree generation is a promising answer to the challenges noted above. In particular, model-based techniques facilitate the integration of available resources into the diagnosis equipment. Furthermore, grounding diagnosis systems on a model provides a systematic way for modification, reuse and optimization.

In our application, model-based approaches have to deal with electrical circuits of the automotive domain. These circuits usually consist of components which show a variety of different behavior types, such as analog, digital, static, dynamic, linear, nonlinear and software-controlled behavior. In principle, model-based techniques provide a systematic way for predicting the behavior of electrical circuits, including faulty behavior. However, adequate modeling of heterogeneous circuits is still a challenge.

In the STILL application scenario, diagnosis follows the branches of a decision tree. Nodes of a decision tree represent fault sets, edges are labeled by the tests (involving measurements, observations, display values and error codes) which must be carried out to verify the corresponding child node. Although the basic concepts of model-based generation of such decision trees are already described in [2] and [5], for the reader's convenience, we briefly outline the main ideas of the approach in the following. Due to the STILL application scenario, we focus on the electrical domain, although, in principle, dealing with devices of different technical domains such as hydraulics or mechanics is feasible.

The first step of model-based decision tree generation is to model a device. This step is supported by a component library and a device model archive (see Figure 1). Design data and knowledge from the design process (knowledge concerning intended device behavior, expected faults, and available measurements) are exploited in this step. In a second step, ok and faulty device behavior is predicted automatically by evaluating the device model. The third step is to build decision trees from behavior predictions. This step is supported by a decision tree archive and a cost model for the tests which can be performed. Decision tree generation can be performed automatically or guided by service know-how, i.e. knowledge concerning preferable decision-tree topologies and fault probabilities.

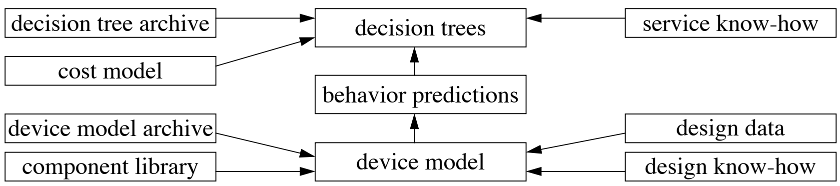


Figure 1. Basic concepts of model-based decision tree generation

In order to realize these concepts, we implemented the MAD system (Modeling, Analyzing and Diagnosing) whose main parts are described in this paper. Section 2 describes MAD's user interface which facilitates adequate device modeling. Furthermore, the internal representation of electrical circuitry is explained. In Section 3, MAD's model-based behavior prediction is described. Section 4 outlines the decision tree generation. The evaluation of the MAD system described in Section 5 was performed in cooperation with the STILL GmbH Hamburg.

2 Device Modeling

In this section, COMEDI, the user interface of the MAD system is presented and MAD's internal representation of electrical circuits is described.

2.1 COMEDI

COMEDI (COmponent Modeling EDItor), the user interface of MAD is similar to a CAD tool which hopefully assures a high degree of acceptance in industry (see Figure 2). For device modeling, COMEDI provides two different libraries, a device model archive and a component library (see Figure 1). The device model archive allows systematic reuse and modification of device models which were created during former modeling sessions. The component library contains different qualitative models of electrical components. A simplified COMEDI model of a forklift frontlight and backlight circuit is shown in Figure 2.

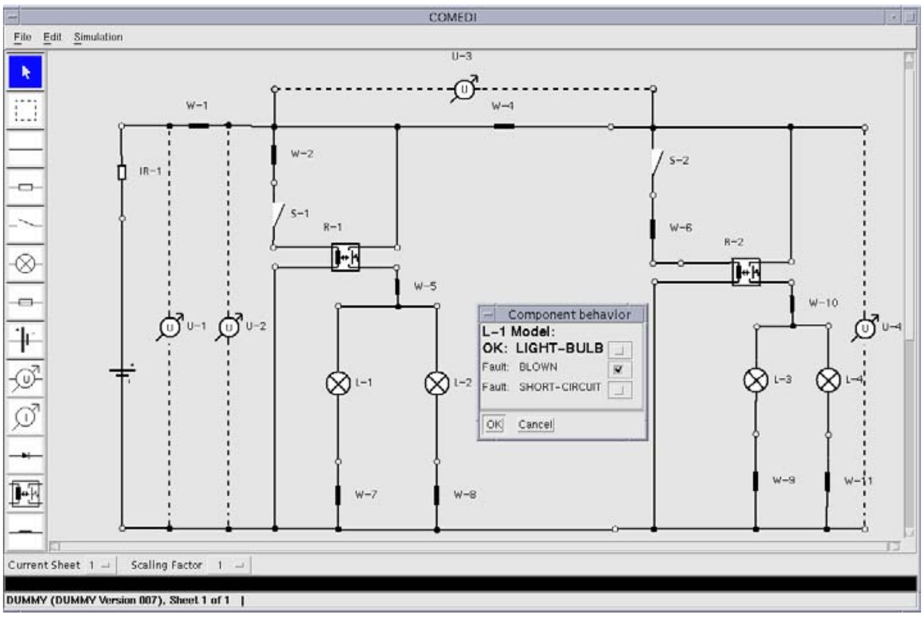


Figure 2. COMEDI model of forklift frontlight and backlight circuit

COMEDI models of simple components represent a single ok behavior mode and optionally several fault behavior modes. Modes of behavior are explicitly marked as correct or faulty. For instance, a COMEDI light bulb model consists of three different behavior modes, i.e. *ok: light-bulb*, *fault: light-bulb-blown*, and *fault: light-bulb-short-circuit* (see Figure 2).

There are electrical components with complex behavior, i.e. components with several operating modes. For instance, hand-operated switches and relays can be open or closed. In COMEDI, each operating mode of a component is described by a distinct model. Each model consists of a single ok behavior mode and (some) fault modes. Additionally, attached to each model, there is a model condition defining requirements under which the corresponding model holds. Alternative models of switches and relays and the corresponding model conditions are shown in Table 1. In this simple example, each component model shows only one faulty behavior mode.

Table 1. Alternative models of switches and relays and corresponding model conditions

component model	ok behavior mode	fault behavior mode	condition
switch model 1	ok: open	fault: stuck-closed	opened-manually
switch model 2	ok: closed	fault: stuck-open	closed-manually
relay model 1	ok: switch-open	fault: switch-stuck-closed	coil-passive
relay model 2	ok: switch-closed	fault: switch-stuck-open	coil-active

Model conditions are of two different types, i.e. internal conditions and input conditions. Input conditions relate to inputs of the investigated technical device. Hence, in order to model certain device input the user of COMEDI can choose the corresponding component model. For instance, in order to model a hand-operated closed switch, a COMEDI user selects switch model 2 (see Table 1).

Internal model conditions relate to internal parameters of the device. For instance, the internal relay model condition *coil-active* (see Table 1) relates to the current through the relay coil. Section 3.2 outlines MAD's automatic behavior prediction and its treatment of alternative behavior models with associated internal conditions.

Since in COMEDI, component behavior is described in a language similar to the way engineer's think about component behavior, design experts can handle the modeling task. This reflects the insight that the design of modern technical systems and of appropriate innovative diagnosis systems is inseparable. In particular, given certain components or subcircuits, knowledge concerning ignorable physical effects as well as know-how about intended behavior is essential to model devices at an adequate degree of abstraction.

Qualitative modeling is adequate because, usually, faults and symptoms are described qualitatively in this domain. Furthermore, qualitative techniques allow to handle parametric variants of a device without changing the device model and, thus, the complexity of diagnosis equipment is reduced. As another point, qualitative modeling reduces the number of fault models because, often, a class of different faults is represented by only one qualitative model. In MAD, dealing with a small number of different fault models is essential because the number of faults determines the size of the decision tree and the computational efforts to generate it. Additionally, using qualitative techniques, electrical circuits can be modeled at an adequate degree of abstraction what is necessary to deal with complex circuits that consist of a large number of components. Thus, for model-based decision tree generation, quantitative network analysis such as SPICE [1] seems to be problematic.

For modeling devices, in COMEDI, component models can be easily combined because of their local internal behavior descriptions (no-function-in-structure principle, [4]) presented in the following subsection.

2.2 Standard Components

Internally, COMEDI models are mapped to formalized standard components showing well defined and idealized behavior. MAD provides four different standard components, i.e. idealized voltage sources, consumers, conductors and barriers. The behavior of idealized voltage sources is well-known from electrical engineering. Consumers are passive and their current/voltage characteristic is monotonous. Idealized conductors do not allow any voltage drop while idealized barriers do not allow any current. Standard components can be connected in combinations of series, parallel, star and delta groupings. This simple internal representation of electrical circuits is sufficient for the following reasons.

- In STILL service workshops, only steady state diagnosis of electrical circuits is performed. Therefore, only steady state behavior of physical components has to be represented in component models. In particular, an explicit representation of temporal dependencies is not necessary.
- A small number of qualitative standard components suffices, because, often, different physical components show similar electrical behavior, i.e. their current/voltage characteristics differ only slightly. Qualitative versions of these current/voltage characteristics are frequently identical.
- MAD's standard components are deliberately selected so that important behavior classes of the application domain can be represented adequately.

Due to analogies between electrics, mechanics and hydraulics, the internal MAD representation is, in principle, also adequate for other technical domains.

2.3 Qualitative Parameter Representation

In electrical circuits, faults may modify component behavior or may even change circuit structures. Hence, heterogeneous symptoms, such as slight deviations of parameter values or total loss of functionality may occur. In general, any circuit behavior that is different from the expected behavior can be a fault symptom. Thus, representing actual parameter values as well as deviations from reference values is helpful to characterize faults and symptoms adequately.

However, MAD's qualitative parameter representation consists of *three* attributes, i.e. actual value, deviation value and reference value. At first sight, this representation may seem to be redundant because *actual value = reference value + deviation* holds. However, for qualitative value spaces this is not necessarily true since a certain qualitative deviation may lead to more than one possible actual value. For example, consider the situation that the reference value of a certain parameter is known to be *positive* and the deviation is *negative*. In this case, the corresponding actual value may be *positive, zero* or *negative*. Hence, qualitative computations can be sharpened if all three attributes are carried along. In [7], we elaborate on this topic.

Table 2 and 3 show attributes and corresponding qualitative value sets of currents and voltages. The semantics of the qualitative values should be obvious. Note that in MAD's internal models of electrical devices, infinite current values may occur because MAD provides idealized voltage sources and idealized conductors as standard components. MAD's set of standard components does not include idealized current sources. Thus, in MAD's internal device models, voltages show certain limits and voltage values beyond these limits can be considered as *impossible* values.

Table 2. Qualitative representation of currents

attributes	qualitative values
actual value / reference value	negative-infinite, negative, zero, positive, positive-infinite
deviation value	negative, zero, positive

Table 3. Qualitative representation of voltages

attributes	qualitative values
actual value / reference value	negative-infinite, negative-impossible, negative-maximum, negative-between, zero, positive-between, positive-maximum, positive-impossible, positive-infinite
deviation value	negative, zero, positive

Due to the MAD's explicit representation of voltage limits, in principle, dealing with logical circuits is possible. For instance, logical values (*low, high)* can be mapped to MAD's voltage values *zero* and *positive-maximum*. Furthermore, MAD's qualitative voltage representation allows to handle electrical devices showing more than one source. In particular, the representation of impossible voltage values paves the way to define a qualitative version of the superposition principle well-known from electrical engineering. Dealing with logical values as well as handling multiple sources is the basis for dealing with hybrid systems consisting of both analog and digital subsystems.

3 Automated Behavior Prediction

In this section, MAD's computation of qualitative values is briefly described. A detailed description can be found in [7]. Furthermore, in this section, MAD's generation of fault-symptom tables is summarized.

3.1 Computation of Qualitative Values

In order to compute qualitative current and voltage values, local propagation methods have been investigated [9]. Since detailed studies proved that local propagation in electrical networks is inappropriate, we follow a different approach first presented in [6]. Networks are transformed into trees representing the network structure. In particular, series, parallel, star and delta groupings are represented explicitly. Exploiting these structure trees, qualitative behavior can in fact be computed by local propagation. Unlike other approaches such as QCAT [8], the SPS method [6], and the Connectivity method [10], MAD offers certain features to improve the accuracy of qualitative prediction. This is explained in the following.

- First of all, rather than relying on qualitative versions of basic arithmetics, MAD computes qualitative values for currents and voltages by a set of qualitative operators which are qualitative versions of complex quantitative equations. In effect, these equations describe behavior of series, parallel, star and delta groupings. Utilization of complex operators avoids multiple applications of simple operators and, thus, avoids spurious predictions. For instance, a voltage divider operator is invoked to compute qualitative voltage values instead of determining current values first and computing voltage values from current values in a second step. In principle, for network analysis, a limited number of operators suffices because MAD's internal representation of electrical circuits offers a limited number of standard components and elementary network structures.

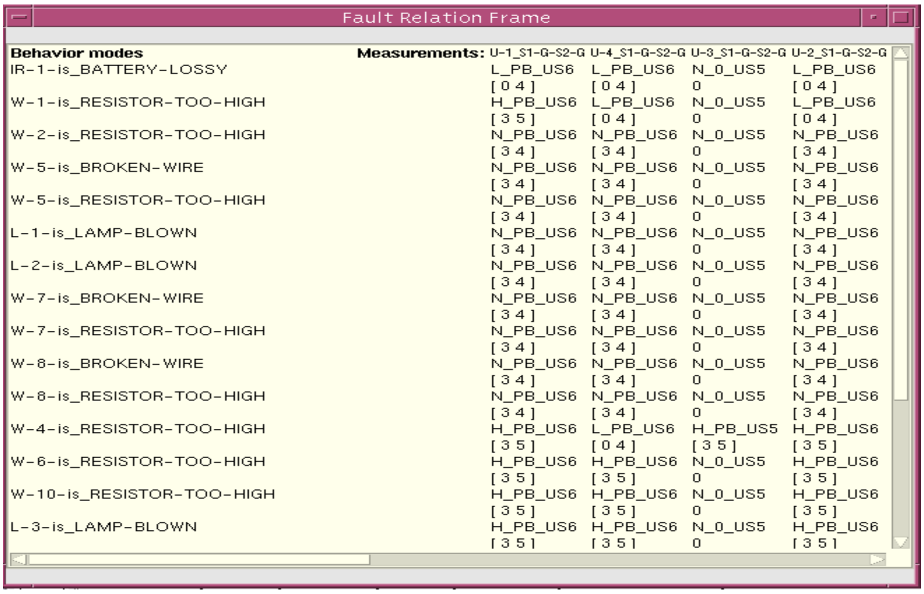
- Second, qualitative operators are defined by applying the corresponding quantitative equation to the interval boundaries which represent actual values and reference values of input parameters. Resulting boundaries represent the corresponding qualitative values of output parameters. Qualitative deviation values are computed from actual and reference values. Additionally, output deviation values are inferred from input deviation values, assuming that parameter dependencies are monotonous. Operators are represented by a set of tables comprising more than 30.000 entries which had to be generated by computer in order to secure reliability. Due to the properties of this qualitative calculus, spurious solutions do not occur at all if the network can be structured into series and parallel groupings of standard components.
- Third, in addition to local propagation of qualitative values, MAD globally analyses network structures and structure trees in order to eliminate (some) spurious predictions. For instance, a global analysis of the network structure allows to determine current directions. Knowledge about current directions can be used to eliminate certain qualitative current values. Therefore, global network analysis may prevent spurious current predictions.

3.2 Dealing with Complex Component Behavior

As stated in Section 2, there are electrical components whose behavior depends on internal parameter values, e.g. the behavior of a relay switch (open / closed) depends on the current through the relay coil. In COMEDI, these components are described by sets of alternative behavior models with associated internal conditions. MAD's dealing with these components is similar to QCAT. In a first step, for each of these components, one of its alternative models is instantiated. Second, qualitative voltage and current values are computed. Third, internal conditions are verified. If an internal condition is violated, one component model is changed and computation of qualitative values is restarted. If all internal conditions are fulfilled, the steady state behavior prediction with the chosen set of models is successful. Steady state behavior prediction fails if all possible combinations of alternative models lead to violated model conditions as may happen in the case of instable behavior. This case is explicitly reported to MAD users.

3.3 Generation of Fault-symptom Tables

In order to generate decision trees, behavior predictions are performed for all operating modes, faults, and fault combination for which diagnosis support is required. For each operating mode and fault assumption, all symptoms (measurements, observations, error codes, display values) are computed which are in principle available for diagnosis. The output of the prediction step is model-based diagnosis knowledge in form of an extensive table of fault-symptom associations. This table is the basis for decision tree generation. For the forklift frontlight and backlight circuit, MAD generates the fault-symptom table shown in Figure 3.


Fault Relation Frame

Behavior modes	Measurements: U-1_S1-G-S2-G	U-4_S1-G-S2-G	U-3_S1-G-S2-G	U-2_S1-G-S2-G
IR-1-is_BATTERY-LOSSY	L_PB_US6 [0 4]	L_PB_US6 [0 4]	N_0_US5 0	L_PB_US6 [0 4]
W-1-is_RESISTOR-TOO-HIGH	H_PB_US6 [3 5]	L_PB_US6 [0 4]	N_0_US5 0	L_PB_US6 [0 4]
W-2-is_RESISTOR-TOO-HIGH	N_PB_US6 [3 4]	N_PB_US6 [3 4]	N_0_US5 0	N_PB_US6 [3 4]
W-5-is_BROKEN-WIRE	N_PB_US6 [3 4]	N_PB_US6 [3 4]	N_0_US5 0	N_PB_US6 [3 4]
W-5-is_RESISTOR-TOO-HIGH	N_PB_US6 [3 4]	N_PB_US6 [3 4]	N_0_US5 0	N_PB_US6 [3 4]
L-1-is_LAMP-BLOWN	N_PB_US6 [3 4]	N_PB_US6 [3 4]	N_0_US5 0	N_PB_US6 [3 4]
L-2-is_LAMP-BLOWN	N_PB_US6 [3 4]	N_PB_US6 [3 4]	N_0_US5 0	N_PB_US6 [3 4]
W-7-is_BROKEN-WIRE	N_PB_US6 [3 4]	N_PB_US6 [3 4]	N_0_US5 0	N_PB_US6 [3 4]
W-7-is_RESISTOR-TOO-HIGH	N_PB_US6 [3 4]	N_PB_US6 [3 4]	N_0_US5 0	N_PB_US6 [3 4]
W-8-is_BROKEN-WIRE	N_PB_US6 [3 4]	N_PB_US6 [3 4]	N_0_US5 0	N_PB_US6 [3 4]
W-8-is_RESISTOR-TOO-HIGH	N_PB_US6 [3 4]	N_PB_US6 [3 4]	N_0_US5 0	N_PB_US6 [3 4]
W-4-is_RESISTOR-TOO-HIGH	H_PB_US6 [3 5]	L_PB_US6 [0 4]	H_PB_US5 [3 5]	H_PB_US6 [3 5]
W-6-is_RESISTOR-TOO-HIGH	H_PB_US6 [3 5]	H_PB_US6 [3 5]	N_0_US5 0	H_PB_US6 [3 5]
W-10-is_RESISTOR-TOO-HIGH	H_PB_US6 [3 5]	H_PB_US6 [3 5]	N_0_US5 0	H_PB_US6 [3 5]
L-3-is_LAMP-BLOWN	H_PB_US6 [3 5]	H_PB_US6 [3 5]	N_0_US5 0	H_PB_US6 [3 5]

Figure 3. Fault-symptom associations for forklift frontlight and backlight circuit

4 Decision Tree Generation

MAD offers three different possibilities to generate decision trees. First, based on fault-symptom tables, decision trees can be created fully automatically. Second, decision trees from archives can be reused. Third, in order to permit manual adaption and modification of decision trees, MAD offers basic editing operations, such as moving a certain fault from one fault set to another and recomputing the corresponding tests. In the following, automated decision tree generation is presented in more detail. One can choose from the following criteria to guide decision tree generation.

- *Minimization of average diagnosis cost*. Automated decision tree generation uses the well-known A*-algorithm [3] to select the tests minimizing the average diagnosis cost according to a cost model which specifies the cost for each test.
- *Grouping by observations, error codes, display values*. Decision trees are generated such that subsets of faults correspond to a prespecified symptom. For instance, all faults are grouped together which cause the frontlights not to shine correctly.
- *Grouping by aggregate structure*. If the aggregate structure of the device is known, decision trees can be generated such that subsets of faults correspond to the same physical aggregate. For instance, faults occurring on a certain board may be grouped together.

Figure 4 shows a decision tree for the forklift frontlight and backlight circuit. This decision tree was generated automatically, guided by the criterion *minimization of average diagnosis cost*. Model-based prediction and automated decision tree generation guarantee, that decision trees are correct and complete with respect to the underlying device model. All faults considered in the device model occur in the generated decision tree, and tests are selected correctly to discriminate fault sets. This holds even if decision trees are modified manually because the editor enforces complete coverage of all faults and correct test assignments. Furthermore, average diagnosis cost is minimal within the constraints imposed by a prespecified decision tree structure.

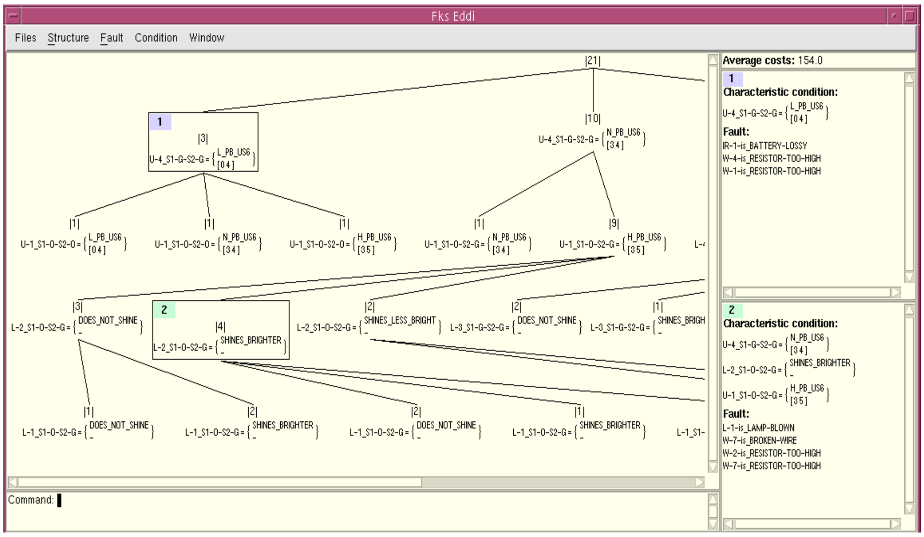


Figure 4. Decision tree for forklift frontlight and backlight circuit

5 Evaluation and Conclusions

The MAD system generates diagnostic decision trees based on a new method for qualitative electrical network analysis which allows accurate behavior predictions for the following reasons. First, MAD's internal standard components represent important behavior types of the electrical domain. Furthermore, since qualitative values describe actual values as well as deviations from reference values, faults and symptoms can be adequately characterized. As another point, exploitation of network structures and certain features to avoid spurious solutions (see Section 2 and 3) assure precise behavior predictions. Using MAD, existing resources such as design data and expert design know-how as well as decision trees and diagnosis knowledge can easily be reused and integrated into decision tree generation.

In cooperation with the STILL GmbH Hamburg, we have evaluated the MAD system in the application scenario and found that using the modeling techniques of MAD with some extensions regarding electronic control units, more than 90% of the faults of the current hand-crafted diagnosis system can be handled successfully. The prototypical implementation allows model-based behavior prediction and automatic generation as well as manual modification of decision trees. Furthermore, we successfully integrated these decision trees into existing STILL diagnosis systems.

Computer-based decision tree generation is a challenging task, because decision trees are wide-spread in industry. With model-based decision tree generation a systematic way for diagnosis system generation has been developed providing the benefits of reduced cost for diagnosis system generation, modification, and maintenance, improved quality management and cost optimal fault identification.

Acknowledgments

We would like to thank our partners in INDIA for their valuable contributions. This work has been supported by the Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) under the grant 01 IN 509 D 0, INDIA.

References

1. Banzhaf, W.: Computer-aided circuit analysis using SPICE, Prentice Hall, Englewood Cliffs, New Jersey (1989)
2. Friedrich, G., Nejdl, W.: Generating Efficient Diagnostic Decision Tree Procedures for Model-Based Reasoning Systems, in: Proc. DX-89, International Workshop on Model-based Systems, Paris, France (1989)
3. Hart, P. E., Nilsson, N. J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths, in: IEEE Transactions on Systems Science and Cybernetics, SSC-4(2): 100 - 107 (1968)
4. de Kleer, J., Brown, J. S.: A Qualitative Physics Based on Confluences, in: AI Journal (1984)
5. Mauss, J.: Analyse kompositionaler Modelle durch Serien-Parallel-Stern Aggregation, DISKI 183, Dissertationen zur Künstlichen Intelligenz (1998)
6. Mauss, J., Neumann, B.: Qualitative Reasoning about Electrical Circuits using Series-Parallel-Star Trees, in: Proc. QR'96, 10th International Workshop on Qualitative Reasoning about Physical Systems (1996)
7. Milde, H., Hotz, L., Kahl, J., Wessel, S.: Qualitative Analysis of Electrical Circuits for Computer-based Diagnostic Decision Tree Generation, submitted to DX-99, 10th International Workshop on Principles of Diagnosis (1999)
8. Pugh, d., Snooke, N.: Dynamic Analysis of Qualitative Circuits, in: Proc. of Annual Reliability and Maintainability Symposium, 37 - 42, IEEE Press (1996)
9. Struss, P.: Problems of Interval-based Qualitative Reasoning, in: Readings in Qualitative Reasoning about Physical Systems, Weld, D., de Kleer, J. (Eds.), Morgan Kaufmann (1990)
10. Struss, P. Malik, A., Sachenbacher, M.: Qualitative Modeling is the Key, in: Proc. DX-95, 6th International Workshop on Principles of Diagnosis (1995)