AST 2009
Workshop on Aviation System Technology

MODEL-BASED DIAGNOSTICS TECHNIQUES FOR AVIONICS APPLICATIONS WITH RODON®

Peter Bunus[1,2], Olle Isaksson[1], Beate Frey[1], Burkhard Münker[1]

[1]Uptime Solutions AB
Ågatan 40, 582 22, Linköping, Sweden

[2]Department of Computer and Information Science
Linköping University, Sweden

peter.bunus@uptimeworld.com

Abstract

In recent years, model-based diagnostics reasoning systems have provided a major advance in fault isolation and reduction of repair time contributing to the reduction of maintenance cost of aerospace and avionics systems. Model-based diagnosis techniques will receive a special attention in this paper and their applicability is illustrated on an aerospace satellite power system from NASA AMES. The paper gives an overview of the model-based diagnosis tool RODON® in which engineering data used in the product development chain is converted into information useful for diagnosis purposes by a process based on mathematical models. We show in what respect the Model Based Diagnostic (MBD), diagnostic trees (DT) are methods of relevance for a complete diagnosis of a system.

1 INTRODUCTION

It is an all too familiar scene from a very famous movie: *Apollo 13*. The movie shows the cockpit of the Apollo 13 where astronaut Jim Lovell (played by Tom Hanks) reports back to earth: "*Houston, we have a problem*". The plot of the movie is based on the third American manned lunar landing mission, part of the Apollo program. Two days after the launch, a malfunction in the spacecraft caused an explosion that made the spacecraft's service module to loose oxygen and electrical power. Astronauts John Swigert, Jr., James Lovell and Fred Haise Jr., who made up the crew of the US's Apollo 13 moon flight, used this phrase to report a life threatening technical problem:

Swigert (LMP):	*Okay, Houston*
Lovell (CDR):	*I believe we've had a problem here.*
Capcom (CC):	*This is Houston. Say again, please.*
CDR:	***Houston, we've had a problem**. We've had a main B bus undervolt.*
CC:	*Roger. Main B undervolt*

LMP: *Okay, Right now, Houston, the voltage is - is looking good. And we had a pretty large bang associated with the CAUTION AND WARNING there. And as I recall, MAIN B was the one that had had an amp spike on it once before.*

Those interested in the transcript of the full technical air-to-ground voice communication of the Apollo 13 mission may wish to consult the internal technical report from NASA 1970 [10].

With increased system complexity in recent years, almost every system under deployment is exposed to component failures or is under the risk of suffering a major breakdown under its lifetime, as the one suffered by Apollo 13. Maintenance and repair is an ever-increasing part of the total cost of a final product. Diagnosis techniques have been adopted by the after market departments of industrial systems product developers as a fast and accurate way of finding the root causes of failures. The shortened development times and the increasing complexity of the products - as indicated by the significantly increasing electrical and electronic content - may lead to difficulties if not handled appropriately. To avoid unnecessary damage, environmental or financial, there is a need to locate and diagnose these faults as fast as possible. This can be done with a diagnostic system, which produces an alarm if there is a fault in the mechanical or electrical system and, if possible, indicates the reason behind it. Traditionally diagnosis is considered the last task in the product design chain. However, the growing importance of lowering the maintenance and repair cost demands for a closer integration of diagnostics tasks in the entire design process and reuse of product related information through all the product development cycle.

In this paper, we present a model-based diagnosis approach for diagnosis of an aerospace satellite power system. However, the approach can be and is successfully being applied to pure aircraft system as well, e.g. to support BITE-development. With the resulting model several standard simulations have been performed in order to calibrate and validate the model. From the developed model, we are able to derive automatically decision trees for troubleshooting of the system in the workshop and decision rules for on board diagnosis. We will also illustrate an interactive model based diagnostics process in which additional measurements are proposed for those situations in which the diagnostic with the initial measurements does not result in a single candidate as a root case.

The rest of the paper is organized as follows: In Section 2, we give a brief description of the principles of model-based diagnosis. In Section 3, we describe the satellite power system from NASA Ames together with the corresponding model used for diagnosis purposes. The model presented in the paper was built using RODON, a commercial model-based reasoning (MBR) system. In Section 3 we illustrate how diagnostics can be performed together with some failure scenarios. We will also illustrate an interactive model-based diagnostics process in which additional measurements are proposed in case that the initial diagnosis does not result in a single candidate as a root case. Then in Section 4 and section 5 respectively, we show how, from the developed model, we are able to derive automatically decision trees for troubleshooting of the system in the workshop, and decision rules for on-board diagnosis. Finally, Section 6 presents a summary of the paper, the future work and our conclusions.

2 PRINCIPLES OF MODEL-BASED DIAGNOSIS

The basic principle of model-based diagnosis consists in comparing the actual behavior of a system, as it is observed, with the predicted behavior of the system given by a corresponding model. A discrepancy between the observed behavior of the real system and the behavior predicted by the model is a clear indication that a failure is present in the system. Diagnosis is

a two-stage process: in the first stage, the error should be detected and located in the model, and in the second stage, an explanation for that error needs to be provided. Diagnoses are usually performed by analyzing the deviations between the nominal (fault free) behavior of the system and the measured or observed behavior of the malfunctioning system.

In Figure 1, a model of a real system (an airplane passenger seat system) is depicted at the lower left corner. It might contain, for example, the behavior of the mechanical components incorporated in the seats, or the behavior of the in-flight entertainment system, or both. Note that, like all models, the model is only an abstraction of the real system (depicted at the upper left corner) and can be incomplete. The granularity of the model and the amount of information and behavior captured into it will directly influence the method employed by the reasoning engine as well as the precision of the diagnostic process.

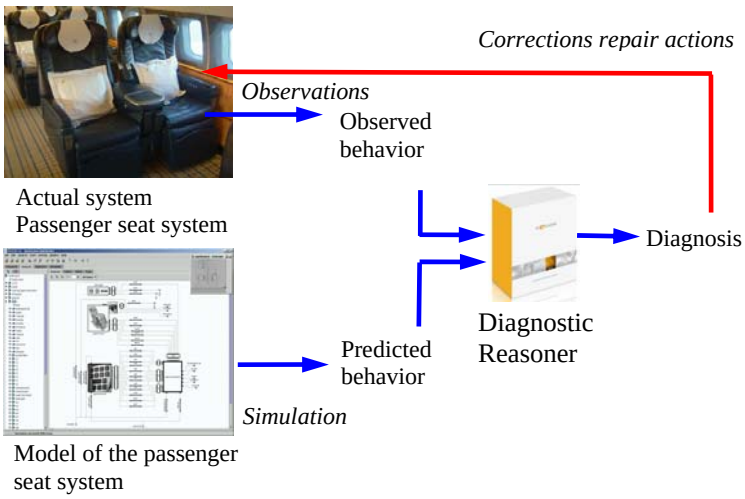


Figure 1. Basic principle of model-based diagnosis

As a general rule, the models are built to enable the identification of failed Line Replaceable Units (LRUs). Once a model of the real system is built, simulation or prediction can be performed on the model. The predicted behavior, which is the result of the simulation, can then be compared with the observed behavior of the real system. This comparison is usually done by a reasoning engine (in our case RODON) that is able to detect discrepancies and also to generate and propose corrective actions that need to be performed on the real system to repair the identified fault. We should note that the process of diagnosis (incorporated in the diagnostic reasoner) is separated from the knowledge about the system under diagnosis (the model). This ensures that the model can be reused for other purposes as well, such as optimization and reliability analysis(Lunde 2003 [5]), model based FMEA support (Zampino and Burow. 2002 [12]) and BITE coverage.

3 THE ADAPT SYSTEM

Assessment and comparison of different diagnostics technologies can be difficult. To facilitate this task the researchers at NASA Ames Research Center have developed the Advanced Diagnostics and Prognostics testbed called ADAPT (Poll, et al. 2007 [11]). The testbed acts as a common platform where different diagnostics tools and technologies, so called test articles, can compete against each other on equal conditions. To achieve this, ADAPT consists of a controlled and monitored environment where faults are injected into the system in a controlled manner and the performance of the test article is carefully monitored. The hardware of the testbed is an electrical power system (EPS) of a space exploration vehicle. The testbed is

located in a laboratory at the NASA Ames Research Center.

3.1 The Advanced Diagnostics and Prognostics Tesbed (ADAPT)

The ADAPT system consists of three major modules: a power generation unit, a power storage unit and a power distribution unit. The interconnections among the different units is depicted in Figure 2 (picture reproduced from Poll, et al. 2007 [11]).

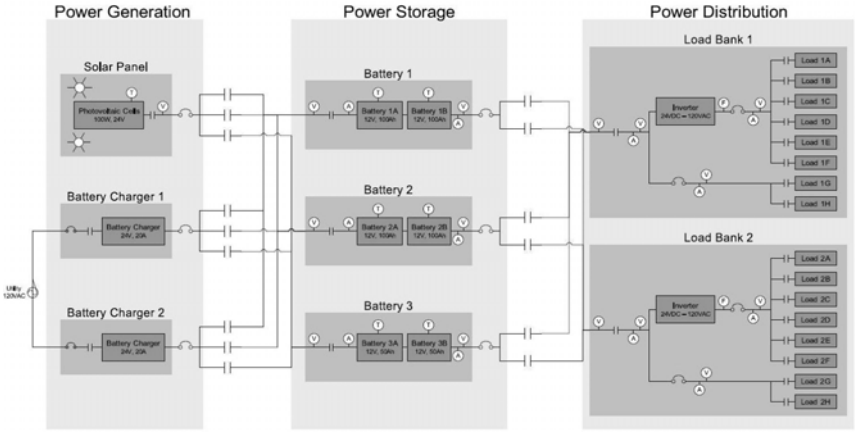


Figure 2. The ADAPT testbed structure (picture reproduced from Poll, et al. 2007 [11])

The *Power Generation* unit can charge the batteries located in the *Power Storage Unit* with the help of two battery chargers and a photovoltaic unit (solar panel). The power generation unit is divided into six subsystems: the solar panel unit, the battery charger panel, the protection and enable panel and three battery-charge selection panels. The power storage unit contains three battery packs and several relays that control the connections between the load bank and the batteries. Circuit breakers protect the power distribution unit from dangerously high currents coming from the batteries. The *Power Storage* unit is divided into two major subsystems: the battery cabinet and the battery-load selection panel. The *Power Distribution* unit consists of two identical load banks. Each load bank is connected to the *Power Storage* unit and powers two DC loads and six AC loads.

A complete description of the ADAPT system can be found in NASA 2006 [9] and Isaksson 2009 [2].

The testbed is controlled by a number of relays and monitored by a large set of sensors. Consequently, it is possible to detect an injected fault and recover from it if the correct action is taken. To facilitate the execution of the experiments performed with the testbed, three operating roles have been defined by Poll, et al. 2007 [11]: *user*, *antagonist* and *observer*. The user simulates an actual crewmember or pilot who operates and maintains the EPS with the help of a vehicle health management application. The antagonist injects faults into the system, either manually by physically acting on the system, or remotely by spoofing sensor values through a computer connected to the system. The malicious actions of the antagonist are not known to the user who is responsible of choosing a suitable recovery action. The observer logs all data in the experiment and monitors how the user responds to the faults injected by the antagonist and therefore measures the effectiveness of the test article. The observer also acts as a safety officer of the experiment and can issue an emergency stop.

Several diagnostics systems such as HYDE (Narasimhan and Brownston [8]), FACT – Fault Adaptive Control Technology (Manders, et al. 2006 [6]) from Vanderbilt University and TEAMS-RT – Testability Engineering and Maintenance System Real Time (Mathur, et al. 1998 [7], Deb, et al. 1998 [1]) have been reported to be integrated and tested on the ADAPT system.

3.2 The RODON model

Based on the complete description of the ADAPT system available in NASA 2006 [9] a corresponding model was built using RODON (RODON is a commercial model-based reasoning (MBR) system developed by Uptime Solutions AB). The model is comprised of 884 components with both nominal and faulty behavior. The model has been restricted to the stationary case and only uses data from one time instance. A dynamic model is under development that will be able to process dynamic input data.

Figure 3 illustrates the higher level of the ADAPT model in RODON with the three main units: the Power Generation unit, Power Storage Unit and the Power Distribution Unit.

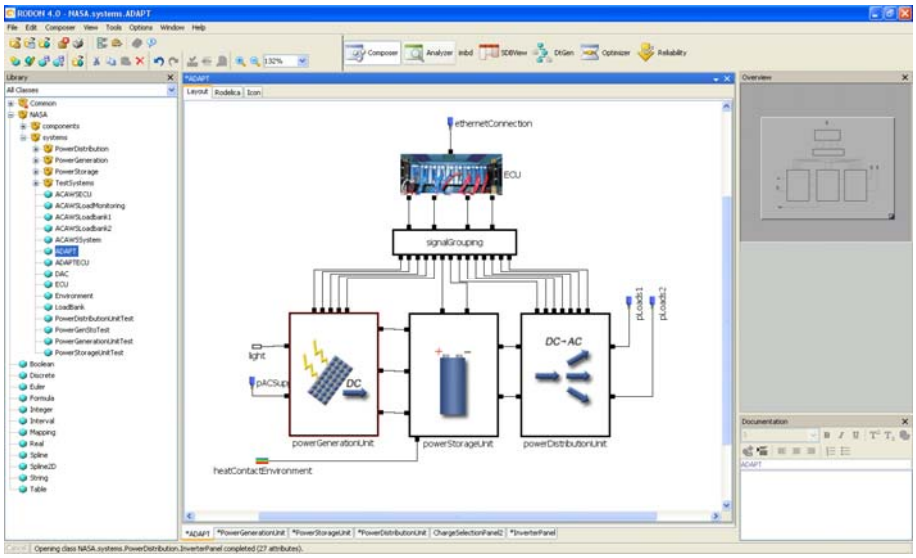


Figure 3. The RODON ADAPT system model showing the three main units: the Power Generation unit, Power Storage Unit and the Power Distribution Unit.

The Power Generation unit, depicted in Figure 4 a) can charge batteries using the energy generated by a solar panel or the by the energy coming from the power grid through a wall socket. The Power generation unit model contains six subsystems: the solar panel unit, the battery charger panel, the protection and enable panel and three battery-charge selection panels. The Power Storage model, depicted in Figure 4 b) has two subsystems: the battery cabinet that contains three battery packs and the battery load selection panel.

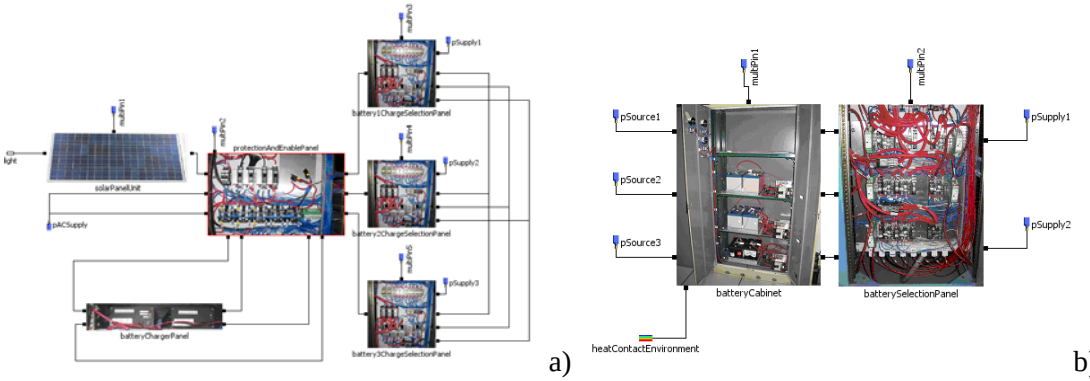
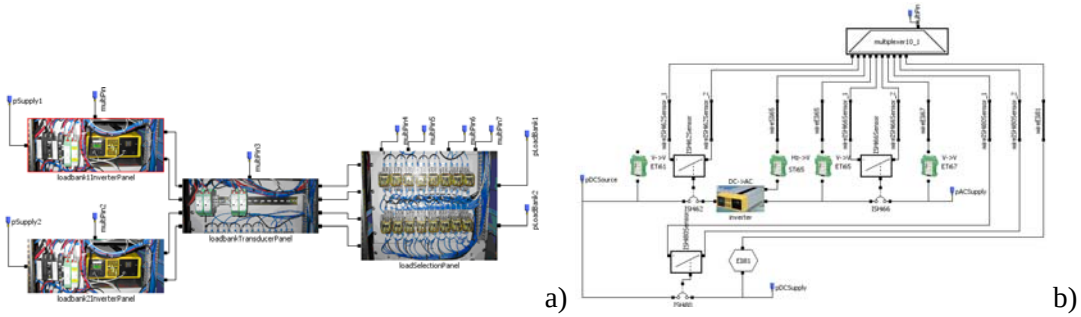


Figure 4. a) The Power Generation unit model. b) High level view of the Power Storage unit.

The power distribution unit consists of two identical load banks. Each load bank is connected to the power storage unit and powers two DC loads and six AC loads. Since the power sup-

plied from the power storage unit is only DC each load bank contains an inverter. The Power Distribution model is depicted in Figure 5 a) and a detailed view of one of the inverter panel subsystems is shown in Figure 5 b).



a) b)

Figure 5. a) The high level view of the Power Distribution Unit b) Detailed view of the Inverter Panel subsystem from the Power Generation Unit.

The implementation details of the power generation, storage and distribution unit models depicted Figure 4 a), Figure 4 b), Figure 5 a) and Figure 5 b) are not relevant for the discussion in the paper. Let us just mention that for each component, the nominal behavior was modeled and augmented with the relevant failure modes, and that variables whose values can be measured in the real system have been marked as observable in the model. A detailed description of those models can be found in Isaksson 2009 [2].

Once the model has been created, RODON supports several diagnostic methods:

- Model-Based Diagnosis (MBD), including interactive MBD, which means that additional measurements can be provided by the user to narrow down the number of diagnostic candidates.
- The automatic generation of decision trees (or diagnostic trouble-shooting trees), which can serve as a model documentation or to assist the mechanic in a workshop in a guided diagnosis.
- The automatic generation of diagnostic rules for on-board diagnostics.

In the following, two of these approaches are illustrated using the model described above.

4 MODEL-BASED DIAGNOSIS WITH RODON

The diagnosis capability of the model has been tested by using 17 different scenarios where faults have been injected into the loads, the power distribution unit or the power storage unit. The scope of scenarios only includes the system from the batteries and downstream to the loads. The scenarios include hardware, software injected faults as well as single, and double faults. Table 1 below lists the fault scenarios together with the corresponding symptoms:

Table 1. Scenarios with the injected faults and the corresponding symptoms

Scenario	Injected fault	Symptom	Candidates generated by RODON	Calc. Time
1a	EY170 stuck open	Lamp box 1 off	**EY170 stuckOpen** wireEC170_1 disconnected wireEC170_1 short_to_gnd wireEC170_2 disconnected	2.9s
1b	ESH 271 disconnected	Position sensor ESH271 report “open”	**ESH 271 disconnected** wireESH271_1 disconnected wireESH271_1 short_to_gnd wireESH271_2 disconnected	3.6s

			wireESH271_2 short_to_gnd	
1c fault1	EY260 stuckOpen	Load bank 2 AC loads off	**EY260 stuckOpen** wireEC260_1 disconnected wireEC260_1 short_to_gnd wireEC260_2 disconnected	2.9s
1c fault2	ESH171 disconnected	Position sensor ESH171 reports "open"	**ESH171 disconnected** wireESH171_1 disconnected wireESH171_1 short_to_gnd wireESH171_2 disconnected wireESH171_2 short_to_gnd	3.1s
1d fault1	EY160 stuckOpen	Load bank 1 AC loads off	**EY160 stuckOpen** wireEC160_1 disconnected wireEC160_1 short_to_gnd wireEC160_2 disconnected	2.9s
1d fault2	EY270 stuckOpen	Pump 1 off	**EY270 stuckOpen** wireEC270_1 disconnected wireEC270_1 short_to_gnd wireEC270_2 disconnected	3.4s
2a	EY141 stuckOpen	Load bank 1 AC loads off	**EY141 stuckOpen** wireEC141_1 disconnected wireEC141_1 short_to_gnd wireEC141_2 disconnected	3.3s
3a	BattA disconnected	Load bank 1 AC loads off	**BattA disconnected** BattA damagedCell BattB disconnected BattB damagedCell batteryCabinet.ground discon.	3.8s
3b	BattB disconnected	Load bank 2 AC loads off	**BattB disconnected** E235 short_to_gnd & E240 short_to_gnd	1'57s
4a	InvA disconnected	Load bank 1 AC loads off	**InvA disconnected** InvA unknown	3.1s
4b	InvB disconnected	Load bank 2 AC loads off	**InvB disconnected** InvB unknown	5.2s
5a fault2	TE500 pinShort & InvB disconnected	Load bank 2 AC loads off	**InvB disconnected** InvB unknown Need dynamic simulation	3.6s
5b fault1	TE502 pinShort & InvB disconnected	Load bank 1 AC loads off	InvB disconnected InvB unknown Need dynamic simulation	4.4s
6a fault1	InvA disconnected	Load bank 1 AC loads off	**InvA disconnected** InvA unknown	6.6s
6a fault1 & fault2	InvA disconnected & ISH180 disconnected	Load bank 1 AC loads off	**InvA disconnected &** **ISH180 disconnected**	2.8s
6b fault1	ISH180 disconnected	Current meter IT181 reports 0A	**ISH180 disconnected**	2.4s
6b fault1 & fault2	ISH180 disconnected & InvA disconnected	Load bank 1 AC loads off	**ISH180 disconnected &** **InvA disconnected**	3.1s

The used data sample is taken after the fault(s) have been injected and before the operator has taken action. The data is taken when most of the transient behavior has settled down. The generated candidates and the calculation times are presented in the last two columns of Table 1. The injected fault is found among the generated candidates in all cases except in the first fault in scenario 5a and the second fault in scenario 5b. The first fault in scenario 5a is very similar to the second fault in scenario 5b: in both cases a fault is injected into a temperature sensor which is monitoring the bulb temperature of a connected lamp. When the lamp is suddenly turned off the bulb temperature is high despite that no power is consumed. This would not have caused any problems if the bulbs are allowed to cool down before diagnosis. In this experiment, the diagnosis was made in a small time window before the operator turned on the backup loads and the lamps had no time to cool down.

4.1 Interactive Model-Based Diagnosis

By using both nominal and faulty behavior of the components RODON it is able to detect single or multiple faults, or to propose additional measurements in an interactive way. Available observations and measurements can be fed to the model in several ways: there is a file interface, a GUI, or a bus interface for direct communication with the real system. As an example, let us consider the scenario in which one of the consumers (an electric bulb from one of the load banks is off while the system is operated under nominal conditions). Obviously, there is a failure in the system that makes the electric bulb off. After entering this symptom into the tool, we can start the model-based diagnostic process to find an explanation for the observed behavior, and to isolate the component that caused that particular behavior. In the first step, the diagnostic engine will compute a list of candidates (hypotheses) that explain the observed behavior:

So far, 18 candidates have been identified (shown on the left hand side of the window depicted in Figure 6) where each candidate corresponds to a single fault which can fully explain the symptom. The list is ordered by the associated confidence values. These confidence values are part of the model and can be imagined as “rough order of magnitude” reliability figures. Components with a lower confidence value are listed first because they are less reliable than others. In the absence of confidence values, the tool will sort the candidates by secondary criteria, for instance lexically. In the graphical user interface, the candidates are highlighted using colour shades ranging from red to blue, with red representing lower confidence value and blue representing less probable candidates.

Dealing with such a big number of candidates is not very efficient in a workshop environment where the mechanic needs to isolate the failure in a very short period of time. There is a need to narrow further down the number of candidates. This can be done by providing extra information to the tool in the form of measurements. The inference engine can profit from this new information to validate the previously computed candidates, and possibly retract those that do not match the measured values. In Figure 6, in the left part of the window, the list of candidates is presented, whereas the right part shows a list of potentially useful measurements or observations to be performed on the system. The latter are ordered by the estimated impact they will have in reducing the number of candidates. The first measurement in the list has the biggest potential to reduce the number of candidates.

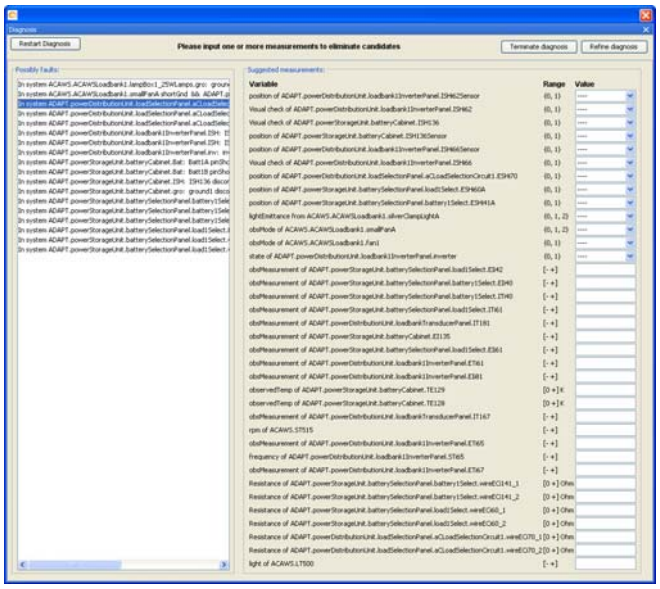


Figure 6. Lists of diagnostic candidates (hypotheses) and of proposed measurements

However, the user is free to choose any measurement from the list. In practice, there might be other selection criteria which are unknown to the tool. For instance, evaluating a fault code activation or a triggered BITE message by a control unit is less expensive than a voltage measurement on a connector block, which involves dismounting one of the units to have access to the electrical part. The diagnostic process can be continued by entering further measurements from the proposed list until the final diagnosis is produced (only one single-fault or one multiple-fault candidate is left). We call this process, in which the user is requested to provide additional measurements to progressively refine the diagnosis, *interactive model-based diagnosis* (IMBD).

5 AUTOMATIC GENERATION OF DECISION TREES FOR TROUBLESHOOTING

In environments where fewer resources are available, a more compact form of diagnostic knowledge representation is desirable. RODON is able to derive several forms of compiled diagnostic knowledge from the model, automatically, by means of a systematic simulation of all essential system states. The modeler has to specify which failure modes and which operational states of the system are relevant for the analysis. The Cartesian product of all those operational states with the set of fault states (plus the state *System ok*) defines a so-called state space. An automatic simulation control module can then be used to simulate each state in the state space, systematically, and to write the results into a data base, which we call *State Data Base* (SDB). The SDB can be used for risk analyses, like failure-modes and effects analysis (FMEA), BITE under-detection or over-detection analysis, and it provides the necessary information for generating decision trees and diagnostic rules.

Decision trees are used to determine which system state explains a symptom, with minimal effort and costs. The root node of a decision tree is the symptom. Leaf nodes are result nodes describing a fault state, e.g. "*w1 is disconnected*". The intermediate nodes are decision nodes which help to discriminate the system state. Decisions may involve a measurement or visual checks to be done by the mechanic. To perform a diagnosis for a selected symptom, the decision tree is traversed starting from the root node, finally arriving at the leaf node with the correct diagnosis. The path through the tree to the diagnosis depends on the answers given at each passed decision node.

The generation process is configurable in a very high degree. In particular, actions required at the decision nodes may be more or less expensive. Consequently, the decision nodes in the generated tree are ordered with respect to a cost measure defined by the modeller. For instance, if BITE/fault code checks are considered to be cheap in comparison to actual measurements, then the resulting tree will ask for evaluation of all helpful BITE messages or fault codes before encountering a decision node in which the maintenance personnel is asked to perform a measurement .

As an example, let us consider scenario 2a in which a fault is injected into the EY141 relay that controls if the first battery is connected to the first load bank. The entire first load bank will lose power. Figure 7 a) show where the fault is injected in the model.

Figure 7 b) shows the automatically generated troubleshooting when that fault symptom is that the power is lost in the entire first load bank. Since the fault was injected by us we know that the failure is due to the fault injected into the EY141 relay. However if a maintenance person would be required to isolate the fault the generated decision tree contains all the necessary steps to correctly isolate the fault. The first node of the tree asks the user to inspect the relay position sensor ESH160A (which reports ”closed”) and then ESH141A (which reports ”open/tripped”). The final leaf contains the correct candidate and the wires leading to the failed relay.

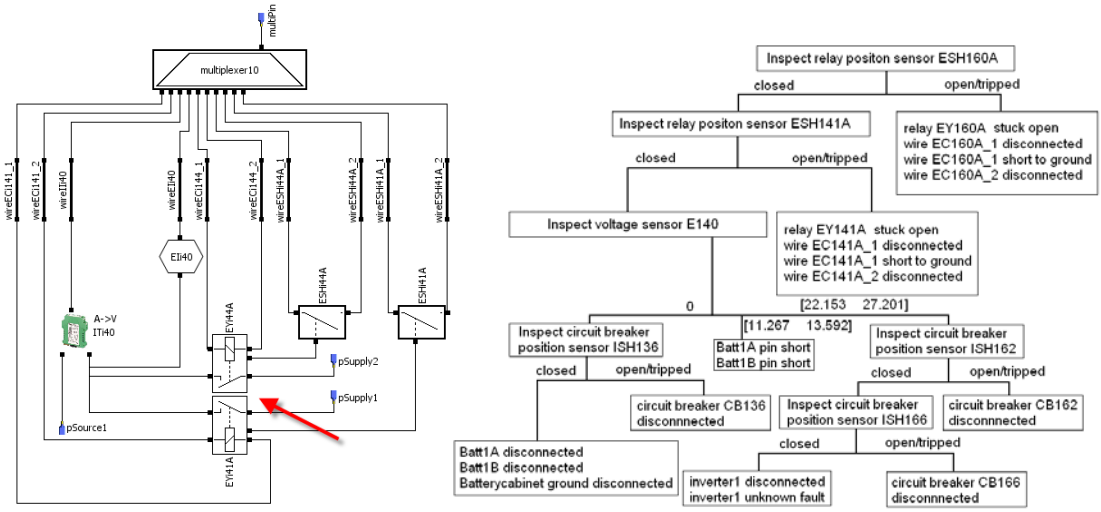


Figure 7. a) A fault is injected into the EY141 relay inside the battery selection unit of the Power Distribution unit causing the entire load bank to loose power. b) Decision tree created for the symptom in which the entire first load bank lost power.

Traditionally, the generation of decision trees is done manually by the system experts, which is an extremely time consuming and error-prone task. Model-based generation of decision trees provides a systematic and safer way to analyze the combinations of all relevant operational states and component failures that can occur in a system, thus serving as a valuable tool in the authoring of troubleshooting documentation. This has been proven in many industrial applications.

6 SUMMARY AND CONLUSIONS

In this paper, we presented a model of an electrical power system (EPS) of a space exploration vehicle. The diagnostics capability of RODON has been demonstrated by applying several fault scenarios to the model for which diagnostics candidates have been generated.

The ability to perform highly reliable diagnostics on a real system reveal its importance in:

- Lowering the repair and maintenance time of the real system which result in lower maintenance costs and increased customer satisfaction.
- Lowering the number of non faulty components that are replaced during maintenance and repair.
- Lowering the downtime and non-operational time of critical systems.

As future work we intend to evaluate the RODON model of this system by running more fault scenarios and experiments and asses the performance of the diagnosis algorithm by computing the benchmarking metrics proposed by Kurtoglu, et al. 2008 [3]. A dynamic model where data from more than one time instance is currently under development. Two simplified versions of the electrical power system (EPS) model will be submitted for the Diagnostics Competition organized by NASA Ames Research Center and PARC hosted at the International Workshop on Principles of Diagnosis DX 2009 (Kurtoglu, et al. 2008 [4]).

Acknowledgements

We would like thank to Werner Seibold and the development team of RODON at Uptime Solutions AB Sweden for valuable discussions and the feedback received for this paper.

7 REFERENCES

[1] Somnath Deb, A. Mathur, P.K. Willett and K.R. Pattipati (1998). Decentralized Real-Time Monitoring and Diagnosis. In Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, USA, 10-14 October 1998

[2] Olle Isaksson (2009). Model-Based Diagnosis of a Satellite Electrical Power System with Rodon. Master Thesis, Department of Electrical Engineering, Linköping University, 2009

[3] Tolga Kurtoglu, Ole Mengshoel and Scott Poll (2008). A Framework for Systematic Benchmarking of Monitoring and Diagnostic Systems. In Proceedings of International Conference on Prognostics and Health Management (PHM 2008). , Denver, CO, USA, 6-9 Oct 2008

[4] Tolga Kurtoglu, Sriram Narasimhan, Scott Poll, David Garcia, Lukas Kuhn, Johan de Kleer and Alexander Feldman (2008). The Diagnostic Challenge Competition 2009 (Dcc'09). Avalilable at https://dashlink.arc.nasa.gov/static/dashlink/media/topic/DXC09_Announcement.pdf, Last accesed 1 February 2009

[5] Karin Lunde (2003). Ensuring System Safety Is More Efficient. Aircraft Engineering and Aerospace Technology, vol: 75, issue: 5. pg 477 - 484, 2003

[6] Eric-J. Manders, Gautam Biswas, Nagabhushan Mahadevan and Gabor Karsai (2006). Component-Oriented Modeling of Hybrid Dynamic Systems Using the Generic Modeling Environment. In Proceedings of Fourth Workshop on Model-Based Development of Computer-Based Systems and Third International Workshop on Model-Based Methodologies for Pervasive and Embedded Software, Potsdam, Germany, March 30 2006

[7] A. Mathur, S. Deb and K.R. Pattipati (1998). Modeling and Real-Time Diagnostics in Teams-Rt. In Proceedings of American Control Conference, 21-26 Jun 1998

[8] S. Narasimhan and L. Brownston Hyde- a General Framework for Stochastic and Hybrid Model-Based Diagnosis. In Proceedings of 18th International Workshop on Principles of Diagnosis, Nashville, TN,

[9] Ames Research Center NASA (2006). Advanced Diagnostics and Prognostics Testbed (Adapt)

System Description, Operations, and Safety Manual. February 2006

[10] National Aeronautics and Space Administration NASA (1970). Apollo 13, Technical Air-to-Ground Voice Transcription. Avalilable at
http://www.jsc.nasa.gov/history/mission_trans/apollo13.htm, Last accesed 31 January 2008

[11] Scott Poll, Ann Patterson-Hine, Joe Camisa, David Garcia, David Hall, Charles Lee, Ole J. Mengshoel, Christian Neukom, David Nishikawa, John Ossenfort, Adam Sweet, Serge Yentus, Indranil Roychoudhury, Matthew Daigle, Gautam Biswas and Xenofon Koutsoukos (2007). Advanced Diagnostics and Prognostics Testbed. In Proceedings of International Workshop on Principles of Diagnosis (DX-07), Nashville, TN, May 2007 2007

[12] Edward J. Zampino and Dirk Burow. (2002). The Application of Rodon to the Fmea of a Microgravity Facility Subsystem. In Proceedings of Annual Reliability and Maintainability Symposium, Seattle, WA, USA, 28-31 Jan 2002