The Use of Function and Component Hierarchies to Generate User Information

Andreas Heinzelmann

Daimler Benz AG, Research FT2/EK T721
70546 Stuttgart, Germany
heinzelm@dbag.stg.daimlerbenz.com

Abstract

This paper describes a technique which allows impaired functions to be automatically determined from the faulty components and vice versa. Moreover a knowledge base containing the criticality of the fault in addition to different abstraction levels. The construction of the hierarchical structure can largely follow automatically because higher variation independent functions can be drawn from a database.

Introduction

Model based diagnosis techniques, for example FR-Dx (Sticklen et al. 1993), GDE (de Kleer and Williams 1992) and it's further development, allow faulty components to be located in a complex system. Other methods, for example Neural Networks (Barschdorff 1991), make it possible to directly generate the fault function of the whole system from measurable signals. For a user display it is necessary to show the faulty or suspect components, the impaired functions and the criticality of the fault. The impaired function must also adapt to the realm of understanding of the individual user. For example, it is necessary to signal the breakdown of a valve in an Anti-lock Braking System (ABS) in a different way to the driver of the vehicle and to the service mechanic at the garage. The provision of such relationships follows mainly via manual assessment of the criticality and user information on the individual components. Partial automatic provision of the relationships between faulty components, impaired functions and the criticality of a fault is, for economic reasons, a forced prerequisite for systems with a variety of variations.

The relationship between the components and the higher partial systems, as well as the relationships between the components and the normal functions connected to the components, can be laid down in a function and component hierarchy. For this purpose a fault tree from the Failure Mode and Effect Analysis (FMEA) can be used (Narayanan and Viswanadham 1987). In the field of diagnosis (Wiedmann 1993), (Abu-Hanna et al. 1991) and (Larsson 1993) lead the way using function and component hierarchies to ascertain the functions of the affected systems in a event of a breakdown. This information is used to display fault details to the user of the diagnosis system. A knowledge acquisition procedure, necessary to generate a component-function relationship for complex systems with many variations, is in practice very difficult.

The following section describes a hierarchical structure to assign fault components to the corresponding functions and vice versa. It is possible to use different levels of abstraction. Different possibilities for the evaluation of such structures will be subsequently explained. The fourth section explains, in general, a process for the automatic generation of the hierarchical structure.

Structure of the Function and Component Hierarchy

A structural hierarchy can be constructed in several way. The hierarchy in this article uses the system designer´s view and is represented by a directed graph split into two sub-graphs (Schweizer 1989). One sub-graph describes the physical structure of the components, i.e. the relationship between the super and sub-systems. The second sub-graph characterizes the relationship between the functions of the whole system and the components. Both graphs are made up of nodes, N_i, and edges, E_j. Edges, E_j, reflect the relationships between the nodes. Components, c_i, functions, f_j, and connecting operations, o_k, make up part of the nodes of the graph. Each node has additional information contained within its characteristics.

The sub-graph reflecting the components' physical structure can be labeled as a compositional hierarchy as it contains nodes. Whether or not the components represent a lowest removable unit (LRU) comes under the nodes' attributes. This should indicate to the service mechanic, that in the event of a breakdown, the device can be removed. Nevertheless, the attribute shows whether or not a node constitutes a superset. Supersets allow the user to group the devices into sets more effectively. The relationship between sub-components or devices to a super one is known as a conjunctive connection. This means that the systems can be clearly assigned to a super-component or device. The orientation of the edges refers to super-components that lie above it.

The second sub-graph describes the function hierarchy. In function graphs the functions are listed as nodes. Every

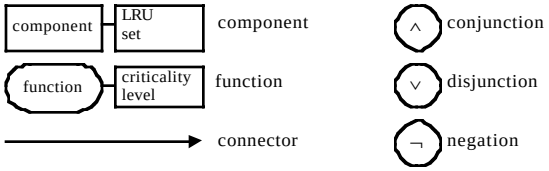

Figure 1. Elements of the Function and Components Hierarchy

node that represents a function contains the level of the task and the criticality as a characteristic of the output information. The level of the task serves as a distinction of the error reports for the different individual user groups, for example, owner or service mechanic. The extent of a breakdown reflects the criticality there of. In addition, the function graph contains connecting-operations, resulting from conjunctive, disjunctive, as well as negation elements. The conjunctive connecting elements do not distinguish between an addition of unrelated sub-functions and a composition of sub-functions. If the upper function has a fault, all unrelated or sub-functions are possible candidates. The disjunctive connecting-operations represent alternative functions, e.g. in safety areas where one can find redundant devices. The negation can handle the loss of a function or a component. For the connection of the nodes the hierarchical function graph contains edges directed towards upper functions. One or more connecting operations can represent the logical dependencies between upper and lower functions. The functions characterize the normal state of the system, while a fault state is described only through the loss of the normal function. In order to connect both of the sub-graphs the whole graph contains edges in which the respective elementary base components are represented by the elementary functions. The elementary functions are the component's functions from the system designer's viewpoint. The function with the relationships implicitly includes the structure between components. Figure 1 shows elements with the basic elements.

The relationships are represented using the example of an ABS-system including a speed display. Figure 2 shows an extract from the principal structure. In each case a speed sensor is used to record the current wheel speed. A software program controls the hydraulic magnet valve for setting the brake power. All these components together make up the ABS function for each wheel. If all four ABS-functions work correctly then the ABS-function has no fault. The necessary speed information for the optimal break effect is made available to another automation unit, via a CAN bus system that indicates the speed of the car for the driver. Alternatively the wheel speed information of the right or the left wheel serves as a speed signal. Figure 3 shows the function and component hierarchy of the ABS example system. Not all characteristics of the individual nodes have been listed for reasons of lucidity.

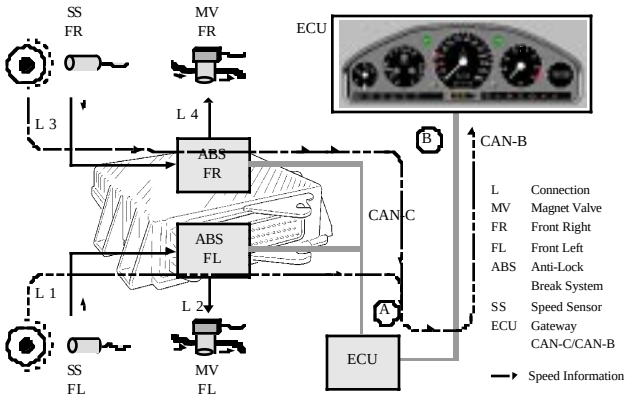

Figure 2 Example System ABS with Speed Signals

Conclusions Drawn From the Hierarchical Structure

The information contained in the representation of the hierarchical system can be used in different ways. One possibility exists that the impaired functions in the corresponding level can be ascertained for a defective component. Another possibility is to begin with a faulty function and determine the component causing the problem. A mixture of both processes is also imaginable. Likewise, the hierarchical structure includes the LRU that corresponds to a faulty component which can be determined by a given function or component.

In order to draw conclusions from the hierarchical representation, dynamic data are necessary to represent the current conditions of the individual components and functions. There are three conditions for the dynamic data:

- normal: The components show no breakdown
- unknown: The condition is unknown
- fault: The component or function is faulty

The relationships expressed in table 1 are used in order to bring the secondary states together with the superior connecting operations with the higher conditions. The relationships describe an output value of a connecting operation with one or two input states. The negation only uses the input I1.

Input States		Output States		
I1	I2	Negation	Disjunction	Conjunction
normal	*normal*	*fault*	*normal*	*normal*
normal	*unknown*	*fault*	*normal*	*unknown*
normal	*fault*	*fault*	*normal*	*fault*
unknown	*unknown*	*unknown*	*unknown*	*unknown*
unknown	*fault*	*unknown*	*unknown*	*fault*
fault	*fault*	*normal*	*fault*	*fault*

Table 1 Bottom-up-Relations

In table 2 the corresponding top-down relationships are listed.

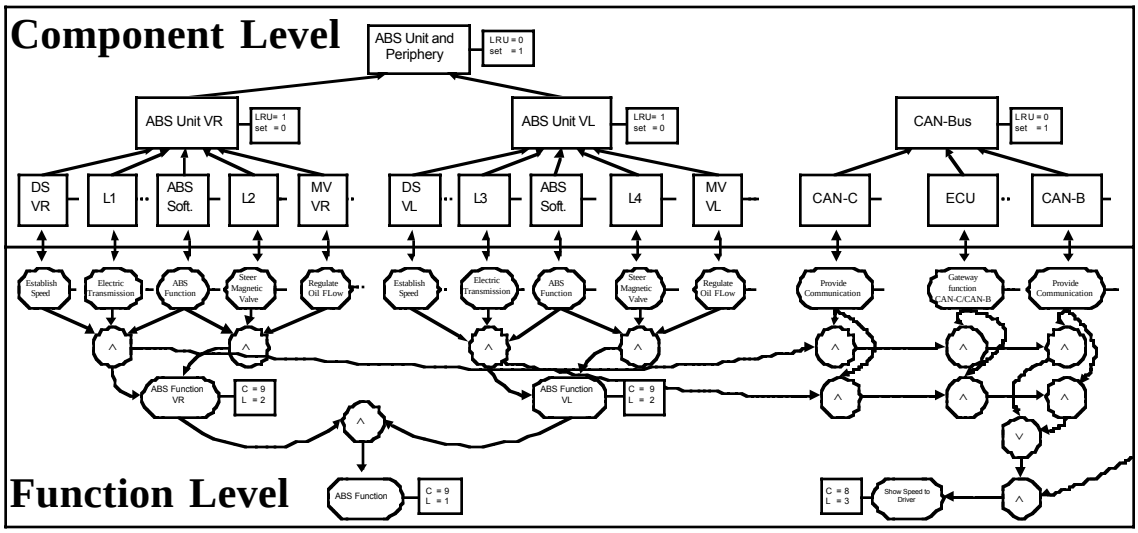

Figure 3 Extract from the Function and Component Hierarchies for the ABS with Speed Signal

Output States	Input States		
	Negation I	Disjunction ∀ I	Conjunction ∀ I
normal	*fault*	*unknown*	*normal*
unknown	*unknown*	*unknown*	*unknown*
fault	*normal*	*fault*	*unknown*

Table 2 Top-Down-Relations

To find the LRU from a component with the condition *fault*, the sub-graph, containing the components, will follow the direction of the edges until the search process discovers a component containing the LRU attribute and the condition *fault*. The conditions of the superior components are shown by the relationships in table 1.

If the conditions of the components are known then a search process starts with the functions' conditions in the lowest level of the function hierarchy. After that the reasoning process searches through the function hierarchy for all functions that display the condition *fault* in the task level selected previously. The reasoning process follows the direction of the edge. The condition of the element during the search process according to the conditions in table 1. The result of the search process is the malfunctioning functions along with the corresponding criticality of the fault.

If the breakdown of a function is given, then the analysis process can determine the components that are responsible. Beginning from the function that has the condition *fault*, the search process determines all subordinate function conditions to the lowest level according to the relationships shown in table 2. The condition of the base functions represents the condition of the components. As mentioned above, the LRU can be calculated from the base components. During the analysis, only those functions and components that demonstrate the condition of *fault* or *unknown* are of interest. A disjunctive connecting-operation during the search process offers an alternative path and thus another possible set of responsible components. Figure 4 shows the possible reasoning process with the input and output data.

In all the above search operations the search process goes through each node and edge a maximum of one time. The time used up is proportional to the sum of the accessible nodes and edges from one starting node. Therefore for a graph that has *n* nodes and *n* edges the maximum steps to ascertain the desired variables is *2n* (O(*n*)).

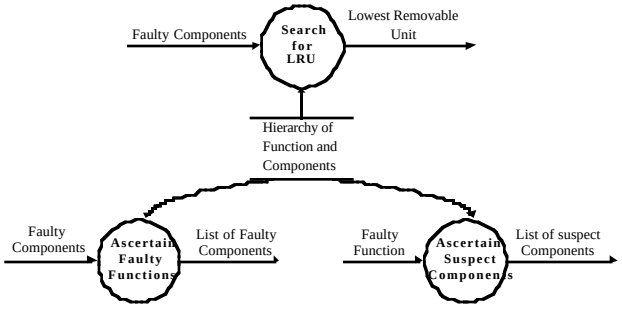

Figure 4 Process of Reasoning in the Function and Component Hierarchy

Generation of the Hierarchical Structure.

For complex systems created from a variety of components, an automatic generation of the function and component hierarchy is practical for economic reasons, especially if a variety of variations of the system are available. An analysis of the underlying knowledge base shows that the function hierarchy becomes more system specific the further down it is followed. Functions that lie on a very

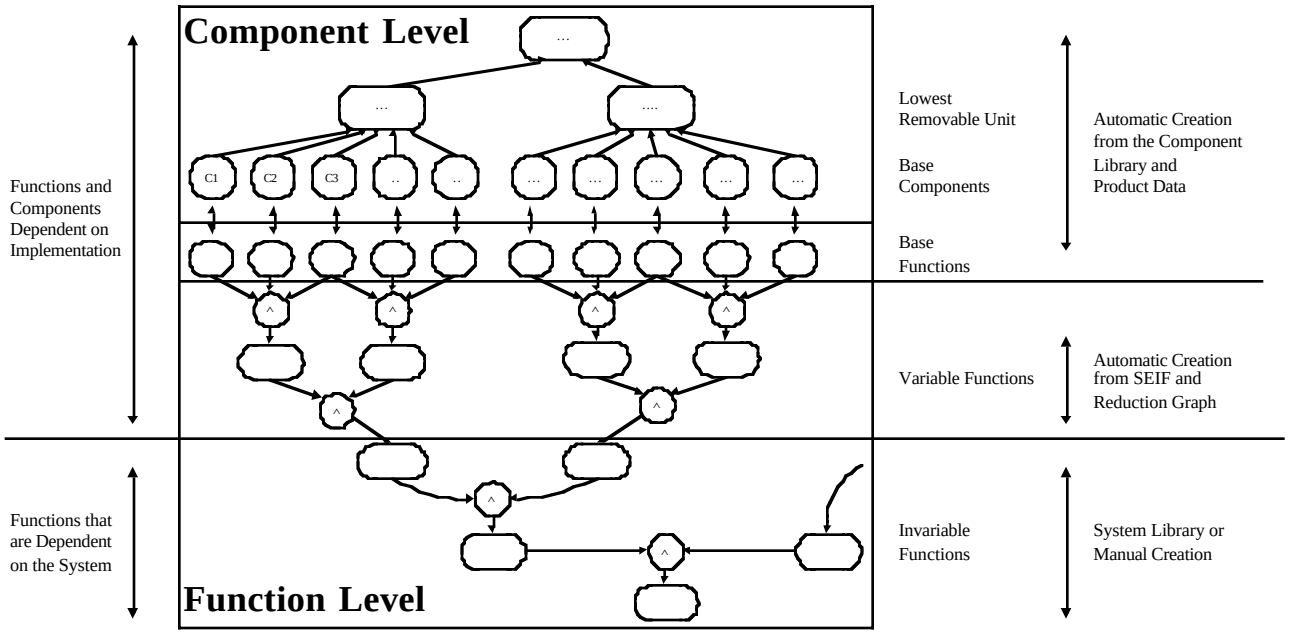

Figure 5 Function and Component Hierarchy

high abstraction level are rarely characterized by their implementation. Functions in the lowest level show a strong implementation dependency that is shaped by individual components. The highest part of the function hierarchy can be seen as independent of variations. Whereas the lower part is dependent on the individual implementation.

An example from the automotive industry presents the two individual parts. Superior functions of a vehicle e.g. *accelerate*, *brake*, *ABS* are available in all variations. The variations differ in their specific implementation. Implementation of the ABS-function can be partially mechanical or electronic. The number of components necessary for the function is dependent not only on the implementation, but also on other variant specific characteristics such as front or rear-wheel driven vehicles. The error messages to the driver are restricted to the driver's knowledge. In contrast, error messages for the service mechanic contain implementation dependent information such as the faulty components.

As the components' attributes can be filed as information in the component library, the component hierarchy is mainly extractable from there. Production data provide the exact number of units that are built into the real system. Grouping of the components can likewise ensue from the component library. Equally the base functions can be filed in the component library. Note that the same component can have different fundamental functions. It depends on the intentions of the system designer. One must take into consideration the intentions of the system designer and his or her specific views regarding each component..

Construction costs for the invariable part of the function hierarchy are limited to the single manual design. The invariable part of the function hierarchy can however be taken from a system library and hardly changes during the life cycle of the product.

In order to generate the function hierarchy that is dependent on the implementation, signal, energy, material and information flows (SEIF) serve as input data. The SEIF can be automatically generated from different simulation and specification tools (Feifel 1998). Each SEIF contains a certain stable function from the whole system. For example in a vehicle there is a signal flow that has the function of presenting the speed on the speedometer. Every component with the corresponding component function makes the speed display possible. The connecting-operations in the function level are represented directly through the signal flow. If there is no SEIF available it is still possible to assign components and their fundamental functions directly to a superior function. Without the availability of SEIFs it becomes practical to directly characterize the superior function through the component. Likewise not all dependency between the components and functions can be described by a directed flow model.

The following conditions are valid for the attributes of the functions in the part that is dependent of the implementation:

- Level of Task: All functions in a SEIF are to be found on the same level. The level of the stable function is always higher than the corresponding SEIFs.
- Criticality: The criticality of the superior function is greater or the same as the criticality of the sub function.

The electrical components present in the periphery of an automation unit can be converted into electrical resistance and condensed into substitute elements. This is possible without losing diagnostic relevant information

(Heinzelmann 1998). The normal state of the electronic components in the periphery of automation units can be represented through substitute elements.

Stable functions are known from the specification of the automation units for every input and output of the automation unit. These functions are assigned to the substitute elements that correspond to the input and output of the automation units. The substitute components with the corresponding function can have a relationship with the real components. In each case a conjunctive connection exists between the components. As a result it is possible to automatically ascertain the functions of the electric components that are located within the periphery and incorporate them into a function hierarchy. These processes can also be transferred, under certain circumstances, to other areas, e.g. the hydraulic system.

Figure 5 shows the regions with the corresponding information from the development process chain. A precondition for the automatic creation is the uniformity of the names given by the different information sources in the development tools. The goal of an automatic creation is that from the beginning of the development of a new product a functional hierarchy exists which contains different references to the specific development and simulation tools. If there is a connection to a FMEA the criticality of faults can be used automatically.

Related work

The difference between the structure described above and the functional system in (Wiedmann 1993) is that here the hierarchical structure can be represented in many logical dependencies. The function hierarchy in (Wiedmann 1993) is connected with the component model on different levels. The connection of the function and component hierarchy in the structure explained above occurs only in the lowest level. This simplifies the automatic generation and the consistency between the function and component structure. In order to reduce the consistency problem (Abu-Hanna et al. 1991) in the description of the system used three levels that were separated from one another with a system of mapping in between. In a fault-tree based structure, e.g. in (Narayanan and Viswanadham 1987), there is no separation between functions and components. The structure explained here is comparable with the FMEA-technology, but strictly distinguishes between function and component. (Lind 1993) and (Larsson 1993) use a similar structure, but the functions are very much device orientated. The structure described here is based primarily on the designer view and upper level functions only have a relation to lower level functions. In addition the MFM (Lind 1993) contains a flow model to describe material, information and signal flows. Furthermore there is no process present in either article that allows for the automatic or semi-automatic creation of a function or component hierarchy. (Thadani and Chandrasekaran 1994) present a matching process for the automatic creation of a function hierarchy with abstract components in the area of electronic circuits.

The decisive difference between the process of inference and the processes quoted here is the universal applicability. In this case components can influence the functions concerned on different abstraction levels. In the same way the components involved can emerge from faulty functions. The structure described in connection with the explained procedures for generated driver information are still under development and are implemented in Microsoft Access™ and C++. It is a part of the research activities in the field of on-board diagnosis.

A further development of the process described could include the top-down conclusion in the condition *suspect component* and produce hypotheses.

References

Abu-Hanna, A.; Benjamins, R.; and Jansweijer, W. 1991. Device Understanding and Modeling for Diagnosis. IEEE Expert, April 1991: 26-32.

Barschdorff, D. 1991. Neural Network Based Condition Monitoring. In Proceedings of the 16th Symposium Aircraft Integrated Monitoring Systems, 586-606. Munich.

de Kleer, J.; Williams, B. C. 1992. Diagnosing Multiple Faults. In: Readings in Model-Based Diagnosis. W. Hamscher, L. Console, J. de Kleer, 100-117, San Mateo CA: Morgan Kaufmann.

Feifel, H. 1998. Automatische Erzeugung und effiziente Repräsentation von funktionalen Modellen zur Onboard-Diagnose in vernetzten Systemen. Dipa. Institut für Automatisierungs- und Softwaretechnik, Stuttgart, Uni..

Heinzelmann, A. 1998. Model-based Fault Simulation, A Reduction Method for the Diagnosis of electrical Component. Forthcoming.

Larsson, J. E. 1993. Reasoning with Explicit Means-End Model. In Working Papers of AAAI Workshop Reasoning About Function, 73-77. Washington, D.C..

Lind, M. 1993. Multilevel Flow Modeling. In Working Papers of AAAI Workshop Reasoning About Function, 86-94. Washington, D.C..

Narayanan, N.H.; and Viswanadham, N.1987. A Methodology for Knowledge Acquisition and Reasoning in Failure Analysis of Systems. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-17, No. 2:274-288.

Schweizer, P. 1989. *Systematische Produktentwicklung mit Mikroelektronik*. Düsseldorf: VDI.

Sticklen, J.; McDowell, J.K.; Hawkins, R.; Hill, T. and Boyer, R. 1993. Troubleshooting Based on a Functional Device Representation: Diagnosing Faults in the External Active Thermal Control System of Space Station FREEDOM. In Working Papers of AAAI Workshop Reasoning About Function, 149-156. Washington D.C..

Thadani, S. and Chandrasekaran, B.1994. Constructing Functional Models of a Device from its Structural Description. In Working Papers of Eighth International Workshop on Qualitative Reasoning about Physical Systems, 276-285. Nara, Japan.

Wiedmann, H. 1993. Objektorientierte Wissensrepräsentation für die modellbasierte Diagnose an Fertigungseinrichtungen. Berlin: Springer.