

Automotive Supplement

Editorial
AutoSteve



Model-Based On-Board Diagnosis and Tools
for the Developer of On-Board Systems
Whole Lifecycle Electrical Design Analysis

MONET is a European Network of Excellence representing Model Based Systems and Qualitative Reasoning (MBS & QR), two branches of Artificial Intelligence Technology. The main aim of the Network is to promote the transfer of these technologies into real life applications for use in Industry.

This supplement to the MONET Newsletter is focused on promoting the aims of the Automotive Task Group. We have brought together information on real world applications of MBS & QR in different sectors of the European Automotive Industry.



Editorial

Welcome to the first in our spotlight series of newsletter supplements on our Task Groups. These supplements will aim to give a clear picture of what the Task Group is working on, its plans for future research and the aims and objectives of the Task Group. This edition is focused on the Automotive Task Group, which is led by Luca Console of the Università degli studi di Torino. There is a short biography of Luca below.

The Monet Project is a European Network of Excellence in Model Based Systems and Qualitative Reasoning (MBS & QR). One of the main aims of the MONET Project is to move the QR and MBS technologies into industry and demonstrate what they can achieve in their fields. The article on AutoSteve is a good example of this as it follows the product from its original design right through to its use in the automotive industry.

Another of the Project's main aims is to promote and disseminate knowledge about the advantages of MBS & QR throughout the European automotive sector, by bringing together experts in these fields from Academia and Industry in order to realise the potential of this area. The methods used in MBS & QR are applicable to many and diverse areas of Research and Industry, including automotive manufacturers, suppliers and also software companies developing control and diagnostic systems. These organisations will transfer outcomes of past, existing and planned European projects throughout the automotive sector and initiate new projects. Their expertise will provide channels for the rapid exchange of experience between researchers working on automotive related developments and automotive systems developers.

We hope to be able to exhibit at some of the forthcoming motor shows and events, so keep an eye out for us and come along and have a chat.



This Task Group is led by Prof. Dr. Luca Console. Luca is currently Full Professor of Computer Science at the Università degli studi di Torino, his interests include Model-based Diagnosis, Abductive Reasoning, Temporal Reasoning and Adaptive Systems.

AutoSteve

This article outlines the development stages that took a concept for a QR Application and saw it realised as a working product. The article is split into three sections. The first section describes how the concept was developed and realised as a working model. The second section outlines how this model was transformed into a commercially viable product and the final section describes how this product has been utilised by Industry in order to solve real world problems.

Academic Inception

(Computer Science Department, The University of Wales, Aberystwyth)

In 1994 an EPSRC project employing 2 Research Associates started the work that would ultimately lead to the AutoSteve design analysis tools and a successful spin off company. Earlier projects had considered the use of model based reasoning for diagnosis, however for the FLAME project we teamed up with engineers at Jaguar cars who were analyzing their sophisticated electrical systems for potential faults. They were using a technique called Failure Mode Effects Analysis (FMEA), which was an ideal application for model based reasoning being an extremely tedious task that required extensive knowledge of the systems being analysed. The project started by attempting to integrate functional, structural and behavioural models with the intuition that the functional model would capture knowledge used by an engineer to interpret vast quantities of results produced by simulation tools. Referring to theoretical work previously carried out in the department by Prof. Mark Lee, a qualitative electrical simulator was implemented in

Pop11. The qualitative simulation allowed easy linking of the simulation results and functions. For example it is easy to link the simulation result "activity in lamp filament resistance" to "light on" as a function. A graphical interface was built to allow each of the three different models to be constructed – and was abandoned shortly after – having realized that the function model was really usefully providing just a set of functional labels. Circuit extracts for each function model were unnecessary because whole system schematics were available and thanks to the speed of the qualitative simulator it was feasible to simulate the entire schematic.

The prototype version of the system was well received by engineers at Jaguar. However, limitations became clear as the structural model could not represent the behaviour of relays or diodes. This led to development of the component builder tool that allowed the structure and behaviour of complex components to be specified by allowing simple behaviour dependencies represented by a simple language.



There was a surprise when one of the Jaguar engineers emailed a component that could not be made to work properly. It was found to contain several hundred lines of dependency statements which was an order of magnitude larger than was ever expected.

Following completion of the FLAME project two follow on projects emerged. A one year commercial quality re-implementation in C++ funded by Ford and also a three year follow up research project called AQUAVIT (Advancing Qualitative Analysis for Verification Interaction and Test). The commercial version was developed at Aberystwyth and was called AutoSteve as both of the engineers with whom we worked most closely were called Steve. The original name – Flame – contained inappropriate connotations according to Ford. The project led to the formation of FirstEarth, a company to support, develop and market the new product. The Aquavit project developed a state chart style representation that allows the specification of more complex components in an intuitive and economic graphical representations; thus solving the excessive complexity problem with the dependency models for some components (by now being used to model Electronic Control Units). Ideas for sneak circuit analysis, a fault tree and a design verification tool were prototyped and subsequently included in the AutoSteve suite by FirstEarth.

FirstEarth has continued development, working with Ford to allow numerical simulation as well as the existing qualitative approach. This is especially useful as a design nears completion and most numerical parameters are known. To distinguish this tool it is known as AQQA to indicate its qualitative and quantitative analysis capabilities.

Two University projects continue the thread of research. The Dougal project is working towards the goal of whole vehicle whole lifecycle design analysis. The interaction between different systems in a modern vehicle has increased the impact and likelihood of intricate failures that occur at system integration. By allowing multiple systems to be analyzed as a complete unit these can be detected and rectified at an early stage. To avoid excessive simulation times work has been done in the area of automatically building abstracted models of systems that can be used efficiently during a simulation whenever no failure exists within the system. Two approaches are being evaluated; one that builds precompiled models of complete system behaviours; one that is able to reuse the results of earlier simulation when a similar situation arises. To enable analysis of a system at all stages of the design process the whole lifecycle part of the project is investigating different levels of simulation granularity, for example qualitative simulation with 5 or 7 values of resistance, combined with automatic reasoning about the results of several analyses carried out as a design evolves and more detailed information becomes available. The notion of function again plays a key role in this work because it provides a common level that allows meaningful comparisons of results to be done automatically.

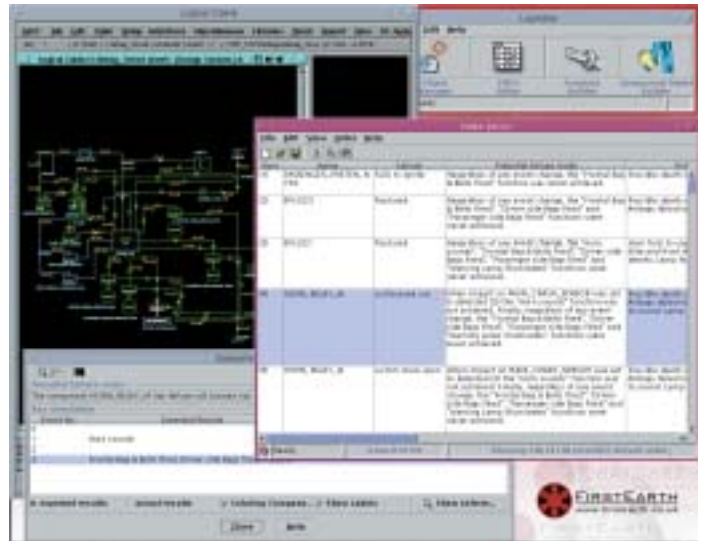
SoftFMEA is tackling the challenges of the newer automotive architectures; networked microprocessors and software replace the traditional relays, switches and isolated electronic or electromechanical subsystems. It is clear that models of the communication methods and network protocols have become important features of the overall system and intersystem behaviour. Work is underway to model these in the same flexible, reusable, and pragmatic spirit that made the original tools powerful and appealing to the engineers that use them.

Commercial Realisation (FirstEarth Ltd)

As the development of FLAME continued it became apparent that there was a potentially large market for such a product. At this time, Ford was changing its tools for electrical design, and they wanted to incorporate the FLAME system within this toolset. At Ford's request, the University of Wales, Aberystwyth (UWA), undertook a yearlong project to redevelop FLAME as a commercial tool that would work with Ford's new ECAD (Electrical CAD) system; this tool became known as AutoSteve.

"We redeveloped the simulation part of the tool in C++, which provided significant speed improvements and it was engineered to compile on a number of different operating systems," explained Neil Taylor, FirstEarth Marketing Director. "The user interface was developed in the scripting language that was part of Ford's new ECAD tool, called PowerView. Adopting this approach, we developed an integrated tool with a familiar user interface for the users, whilst providing the power of C++ for the underlying simulations."

Shortly after delivering the first version of AutoSteve, FirstEarth was established to exploit this new commercial application. It became clear that the automotive industry uses a number of different ECAD tools to design electrical schematics. The decision was taken to redevelop the user interface part of AutoSteve in the Java programming language. This provided a powerful cross-platform



Engineers can interactively investigate the FMEA reports that are automatically generated by AutoSteve. The results of each row in the report can be visualised through back colouring onto the schematic.

user interface that would enable AutoSteve to be adapted to work with a wider number of ECAD systems. This decision has paid off, with Volkswagen's advanced Research and Development group selecting a version to work with the SaberSketch tool. AutoSteve also works with tools from Mentor Graphics and VeSys.

AutoSteve provides automated FMEA and Sneak Circuit Analysis (SCA) tools in one package. FirstEarth has also released a tool specifically for SCA, called SneakExplorer. The tools have mainly been used in the automotive industry, at Original Equipment Manufacturer (OEM) and Tier-1 suppliers. The techniques and model-based technology that underpins the simulation is more generally applicable to electrical and electronic systems



development. "We are getting enquiries from several different industries, including aerospace and defence," said Mr Taylor. "As the complexity of electrical systems increase, companies are looking for intelligent tools that help them analyse their products' safety and reliability."

FirstEarth is actively developing new releases of the AutoSteve and SneakExplorer tools. The development programme is a mixture of new features, including the integration of recent research into the area of model-based reasoning from UWA, customising the product to meet customer requirements, and to extend the types of analysis and re-use of the results within the customer's product lifecycle.

"This is an exciting time for electrical design analysis," said Mr. Taylor. "The model-based reasoning is enabling companies to perform their analysis significantly earlier than traditional manual approaches providing major cost savings. There is also the ability to use the results of the FMEA/SCA later in the product lifecycle. Applications include developing test-board scripts and re-using the FMEA results to help automate the development of diagnostic system."

Industrial Application

(Ford Motor Company)

"Safety and Reliability are very important to Ford," explains Andreas Kraus, Manager, Electrical Technologies & CAE (Europe) of the Ford Motor Company. "Automobile electrical systems are getting more and more complex, and we need to carry out thorough FMEA (Failure Modes and Effects Analysis) and SCA (Sneak Circuit Analysis) for each product, while reducing our product development times."

FirstEarth helped the Ford Motor Company to make the improvements they needed. The AutoSteve™ design analysis tools generate comprehensive FMEAs in minutes, freeing up weeks of skilled engineer time and reducing the number of breadboards and physical prototypes required.

AutoSteve is designed to allow close integration with a number of different Electrical Computer Aided Design (ECAD) tools. Through the integration of AutoSteve with Ford's standard ECAD tool, TransCable™ from Innoveda, Ford benefits from an advanced design analysis solution for the electrical design department, without having to change the way they work.

FMEA is the process of analyzing how individual component failures affect the overall functionality of a system. For example, if a relay's coil burns out, what happens in an airbag circuit when the electronic control unit detects a crash?



The latest Mondeo



Ford engineers receive their European Technical Achievement Awards at a presentation lunch.

© 2001, Ford Motor Company

Ford say "FirstEarth tools save time and money"

The output of an FMEA is a report that details the effects caused by each possible failure in the design. Each effect is assigned a Risk Priority Number; a larger number indicates a higher risk. Ford engineers analyze the report and decide what action may be required to address the issues raised, for example a partial re-design.

SCA involves identifying unexpected interactions in an electrical circuit. A common cause of Sneak Circuits is a combination of switch positions that wasn't originally considered by the designer and that may have unexpected or dangerous side effects.

A traditional, manual FMEA involves working meticulously through the implications of a particular component failing. The work is time-consuming and often tedious, but it needs the skill and experience of scarce design analysis engineers. Until now.

FirstEarth has developed AutoSteve, a program that simulates all possible activity of a circuit and automatically writes an FMEA report detailing what would happen if any component failures occur. Using the same simulation technology, FirstEarth developed a solution to automate SCA.

Ford has adopted AutoSteve as part of its global – engineering – strategy. It is in use at centers in Detroit (USA), Dunton (UK) and Cologne (Germany).

"When we performed an FMEA by hand, we would form a team that would typically spend months on a detailed analysis. The analyses were not exhaustive and could contain inconsistencies. We wanted a way to reduce the time spent on our design, analysis, in particular FMEA and SCA, and improve the coverage and quality," says Mr. Kraus. "AutoSteve has given us everything we were looking for. We are able to experiment with more designs and to analyze the safety and reliability consequences of design alternatives, whilst reducing our overall design time and costs. AutoSteve paid for itself the very first time we used it on a production project."

Ford has made additional savings, because more thorough analysis means lower warranty costs and fewer recalls.

"AutoSteve increases quality, reduces risk, and saves money, while giving us a faster time-to-market," says Mr. Kraus. "AutoSteve is making a contribution in these areas for Ford."



Model-based On-Board Diagnosis and Tools for the Developer of On-Board Systems – VMBD and IDD: Two Projects of the European Car Industries

P. Struss

Technical Univ. of Munich, Computer Science Dept., Orleansstr. 34, D-81667 Munich, Germany and

OCC'M Software GmbH Gleißentalstr. 22, D-82041 Deisenhofen, Germany
struss@in.tum.de, struss@occm.de

ABSTRACT

The growing importance of on-board diagnosis for automobiles demands for new diagnostic methodologies and techniques and for a close integration of diagnostic tasks in the entire design process. This report describes work carried out within two European projects. In the "Vehicle Model based Diagnosis" (VMBD) project, demonstrator vehicles with built-in faults provided a serious challenge to model-based diagnosis techniques and a real-life test-bed for their evaluation. One of the guiding applications within VMBD was model-based on-board diagnosis of faults in a turbo diesel engine system with a focus on potential origins of increased carbon emissions. The second, still on-going, project, "Integrated Design Process for onboard Diagnosis" (IDD), developed a model of a new design process which allows for a better integration of diagnosis related tasks, such as diagnosability analysis, failure-modes-and-effects analysis (FMEA), on-board diagnosis design, in the overall design process of mechatronic subsystems.

INTRODUCTION

Research on model-based diagnosis (e.g. [Hamscher et al. 92], [Dressler and Struss 96]) has generated a number of well-founded theories and sophisticated prototypes of implemented diagnosis engines. However, many of these diagnosis systems have only been applied to toy examples or to problems that ignored the practical context of industrial applications. As a result, the transfer of the technology into practice is well behind the expectations, despite the fact that it promises to meet some crucial requirements of automated diagnosis for industrial needs.

Car industries provide a good example of such industrial needs. It is estimated that European passenger cars have an average yearly downtime of 16 working hours due to malfunctions and maintenance. This figure is even greater for commercial vehicles. For the European Community alone, this amounts to a total of over one billion hours for diagnosis and repair. At the same time, with increased environmental awareness, stricter constraints are imposed on the car manufacturers to develop clean cars, and also to keep them clean during their life cycle (see, for example, [OBD 93]). These growing constraints are reflected in increased requirements on on-board diagnostics development. For engine management control units, currently about one half of the software is dedicated to diagnosis, and this share is still growing.

This contribution presents work on transferring model-based systems technology to industrial practice in order to provide a new methodology and new software solutions that are required to address the needs for both reliable and efficient diagnostics of vehicles and systematic and economic processes for generating them.

We first describe the realization of a prototype of a model-based on-board diagnosis system within the Brite-EuRam project VMBD (Vehicle Model Based Diagnosis) and its theoretical and technical foundations. Next, we present the objectives and intermediate results of the ongoing European Fifth Framework project "Integrated Design Process for onboard Diagnosis" (IDD) which aims at developing tools for the designers of on-board systems.

THE VMBD PROJECT

The Brite-EuRam joined several car manufacturers and suppliers project VMBD (Vehicle Model Based Diagnosis) with the intention to promote the transfer of model-based diagnosis technology by the challenge of applying it to on-board and off-board diagnosis of passenger cars. The results and system performance were evaluated on real demonstrator vehicles. Within this project, Volvo Car Corporation, Robert Bosch GmbH, and OCC'M Software GmbH produced a model-based system that diagnoses problems related to increased carbon emissions of diesel engines, a problem of significant importance with respect to environmental impact and compliance with legal requirements. The system transforms the sensor signals that are available to the standard electronic control unit (ECU) on-board to a qualitative level and exploits them for detecting and localizing faults based on a model of the turbo control system. It has been installed on a Volvo demonstrator vehicle with a number of built-in faults.

Figure 1 shows the part of the system which is responsible for supplying air to the diesel engine. It can be decomposed into the exhaust gas recirculation (EGR) subsystem (upper part of Figure 1) and the turbo control subsystem (lower part of Figure 1).

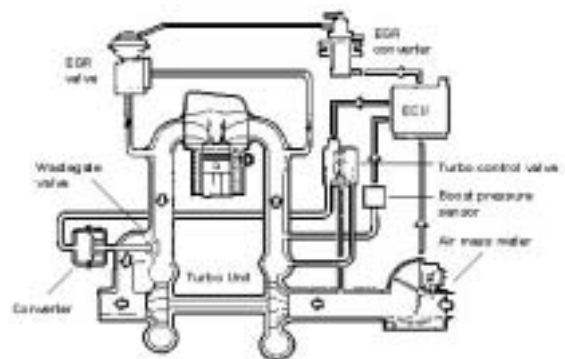


Figure 1: Turbo control and exhaust gas recirculation subsystem of the DTI

The purpose of the exhaust gas re-circulation system is to return a certain amount of the exhaust gas to the intake air to decrease the oxygen rate of the intake air and thus to reduce emission levels of the fuel combustion. Depending on driving conditions, the ECU governs the EGR converter to achieve a certain air pressure in a control pipe, which in turn sets the position of the exhaust gas re-circulation valve. The position of this re-circulation valve then determines how much of the exhaust gas is fed back to the air intake pipe.

The turbo control subsystem consists of a turbo-charger turbine, which is driven by the engine's exhaust gas, for compressing (and thereby increasing the mass of) the air taken into the engine. The ECU controls the boost pressure (i.e. the pressure in the engine intake pipe) admitted in a certain driving situation by opening or closing the turbo control valve, which determines the position of a so-called waste-gate valve.



The position of this valve determines how much of the exhaust gas drives the exhaust turbine of the turbo-charger.

The ECU not only issues commands to the actuators, but also monitors and checks the sensor values it receives from these systems. The goal of this so-called on-board diagnosis is to signal alarms to warn the driver in case of a failure and to generate fault codes that can be further used in the service bays to track down a failure.

In accordance with the overall thrust of the project, our goal thus was

- to produce a prototypical model-based diagnosis system that is capable of diagnosing faults in the diesel engine based on the sensor signals that are available to the ordinary ECU,
- to this end, generate a library of models of the relevant components, and
- to perform this task in a systematic way as a contribution to a general methodology for producing on-board diagnostics.

In the following, we briefly outline the key idea of the approach to diagnosis and summarize the result of the experiments. For more details, we refer to [Sachenbacher-Struss-Weber 00].

CONSISTENCY-BASED DIAGNOSIS TECHNIQUES

In a nutshell, the standard, so-called consistency-based approach to diagnosis ([de Kleer-Mackworth-Reiter 92], [Dressler-Struss 96]) can be described as follows (see **Figure 2**):

- Observations of the actual behavior of the system are entered.
- Based on the device model, conclusions are computed about system parameters and variables (observed and unobserved). For each derived prediction, the set of component models involved in it is recorded. This information can be determined by the diagnosis system because the device model has a structure that reflects the device constituents.
- If a contradiction is detected, i.e. conflicting conclusions for a variable occur (fault detection), the set of components involved in it indicates which components possibly deviate from their intended behavior.
- Diagnosis hypotheses are generated, i.e. sets of faulty components that account for all detected contradictions (fault localization).
- In case models of faulty behavior are provided, the same approach (checking consistency of a model with the observations) can be used to discard particular faults (fault identification) or to conclude correctness of certain components if the set of modeled faults is considered complete.

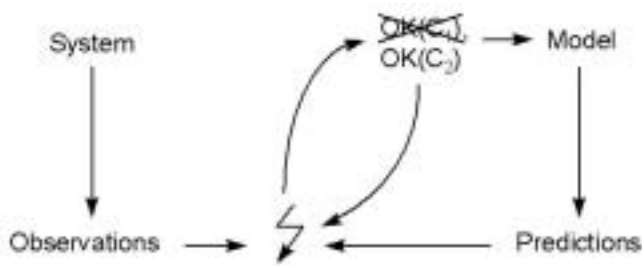


Figure 2: Consistency-based approach to diagnosis

This diagnosis framework has the desired property to systematically exploit the analytic redundancy among the available sensor signals. The model-based approach alone provides one answer to the methodological challenge, because its underlying principles (and the implementation) are independent of the particular subsystem and enable the re-use of the involved software components. Generating a specific diagnostic system is thus reduced to generating an appropriate model of the system to be diagnosed.

As stated above, component-oriented modeling is a natural approach in our application domain. Beyond this, it is the key to solving the variant problem, because the model of a subsystem is derived as the aggregation of standard building blocks. This is another element of a general methodology and enables the automated generation of a device model and, hence, of a tailored diagnosis system based on a structural description of the device only (which should be the natural output of a CAD system). A way of creating diagnostics for all variants of vehicle subsystems is thus obtained that is systematic as well as efficiently supported by computer tools. **Figure 3** illustrates this idea.

For diagnostic purposes, faults can be described as certain component failures, and fault models associated with the respective components. This provides a principled way of capturing knowledge about faults in a modular way which contrasts other approaches in AI (based on storing associations between symptoms and faults for each device in terms of rules or cases) or engineering (trying to identify parameter deviations in a closed mathematical model of the entire device).

Since a component model is meant to be used within the contexts of various devices, it has to capture a behavior description which must not presume a specific context and, particularly, not the correct functioning of the rest of the device. The strict discipline in modeling required to achieve this goal is another important element of the methodology.

It is interesting to note that we need not to build a model of the control unit behavior itself, unless we want to detect faults in the ECU. Due to the fact that the model runs within an on-board environment, all the control unit's signals will be available for observation. Consequently, a behavior model of the control unit could never be part of a diagnostic hypothesis, and would therefore be useless.

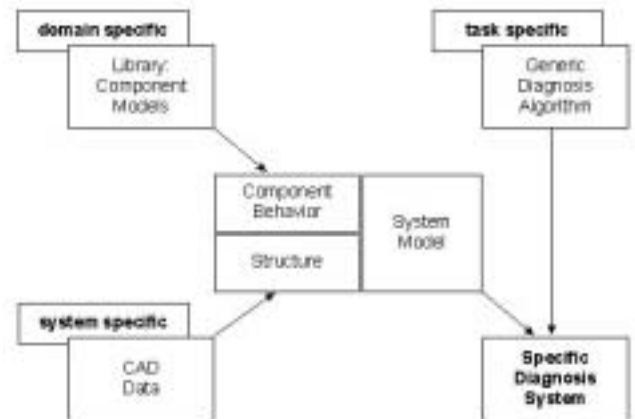


Figure 3: Automated generation of model-based diagnostic systems

EVALUATION ON THE DEMONSTRATOR VEHICLE

The software for the on-board diagnosis prototype consists of two components:

- a module for the conversion of raw signals into qualitative observations, and
- a model-based run-time system that performs diagnosis on the basis of these observations.

This was based on components of the commercial RAZ'R system of OCC'M Software that offers a development environment for diagnostic models as well as a run-time version of a consistency-based diagnosis engine ([RAZ'R 02]).



In the VMBD project, a Volvo 850 TDI demonstrator car was made available for hands-on experimentation with the DTI application. Failures can be induced in the car during various operational conditions of the engine with the model-based diagnosis system running, and the results can be compared with the conventional diagnostic capabilities of the control unit. The various failures in the demonstrator car can be adjusted by potentiometers and triggered by switchboards from inside the passenger compartment (see **Figure 4**). A pneumatic leakage, for example, is simulated by additional valves opened and closed by electrical switches.



Figure 4: View of the Volvo Demonstrator Car showing the notebook connected to the ECU. The glove compartment (behind) contains the switchboard for controlling the built-in faults.

We were particularly interested in failures that were not captured by existing on-board diagnostics. Since increased legislative and customer demands have led to new requirements especially for aspects related to emissions and performance of the system in the Volvo car, we concentrated on effects that involve incomplete fuel combustion and increased carbon emissions due to an excessive quantity of fuel injected or insufficient airflow to the engine (called "black smoke" problems). Types of failures which can lead to black smoke symptoms involve leakages in pipes, malfunctions of valves (e.g. stuck-at-open or stuck-at-closed), increased friction in bearings (resulting in a delay of actuators) or signal disturbances due to electrical failures.

One scenario in the demonstrator car consists of a leakage in the air hose between the turbine outlet and the engine intake manifold. The scenario was realized in the car by installing an electric motor which opens a valve to release pressure from the inter-cooler system via a 12mm opening. If the leakage is opened, air (oxygen) mass is lost after having passed the air mass sensor. The fuel quantity calculated by the control unit which is based on this signal will therefore be too high for the actual amount of oxygen in the combustion chamber. This leads to incomplete combustion of the diesel fuel, which causes increased carbon emissions in the exhaust gas (due to non-burnt particles) and reduces the torque of the engine. This effect is, depending on the driving condition, perceivable for the driver as black smoke emerging from the exhaust system.

From the available control unit data, the following subset of signals was fed to the prototype for diagnosing the described scenarios:

- atmospheric pressure sensor signal
- boost pressure sensor signal
- mass airflow sensor signal
- engine speed sensor signal
- duty cycle of the turbo control valve
- current fuel quantity injected.

The on-board diagnosis prototype uses only these control unit signals, and no further signals from additional sensors. The current control unit software in the turbo control system is not able to detect any of the above failures based on the same signals. The frequency at which the control unit reads the signals from the sensors varies with the speed of the engine, therefore the time points at which observations occur are not evenly distributed.

Figure 5 shows the diagnostic results for a slowly opening leakage during stalling the engine. The upper part of the window shows the control unit signals listed above. The measurement runs for 9.75 seconds and yields 1064 quantitative observation vectors. The signal transformation module reduces them to only 12 qualitative observation vectors (indicated by the small "peaks" at the base of the signal window).

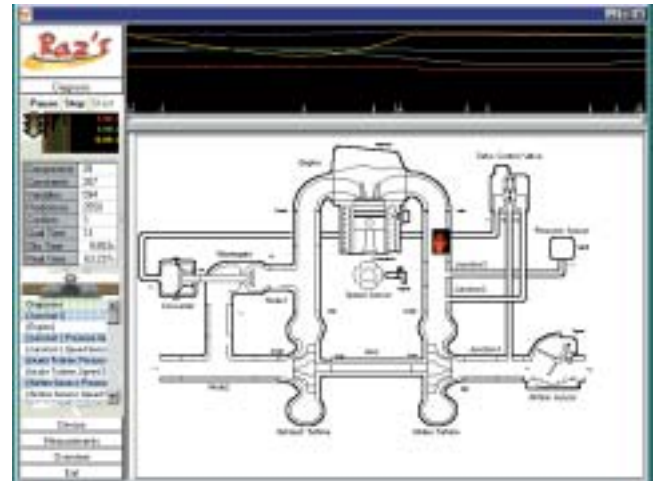


Figure 5: Screenshot of the model-based run-time diagnosis prototype for the DTI turbo control subsystem

The two single fault hypotheses generated by the system contain the component where the failure was actually induced ("Junction3", see mark in Figure 5 within the window depicting the system structure).

The runtime for the example (on a Windows/Pentium PC) is 25,85 seconds without using temporal caching, and 2,87 seconds if temporal caching is activated. This means that, for this example, the performance of the on-board system is in the order of magnitude of real-time.

In summary, this demonstrator, like two others produced in VMBD (see [Cascio et al. 99], [Bidian et al. 99]), successfully proved the feasibility of model-based techniques for on-board diagnostics of vehicles.

THE IDD PROJECT

The results of the VMBD project triggered the interest of the European car industry in further steps towards the introduction of the technology into industrial practice. This requires to refer to the actual industrial work processes related to the design and implementation of on-board diagnostics.

At present, there is no correspondence between the important role of diagnosis in onboard systems and a similar role that diagnosis should play in the design process chain.

The European Fifth Framework project "Integrated Design Process for onboard Diagnosis" (IDD) pursues the goal to formalize and standardize the diagnostic design process, and to enable the introduction of diagnosis early in the chain. This methodological goal has to be combined with another important objective: *giving to the designers a set of model-based tools that can help them in evaluating and understanding the effects of each choice on the system being designed*. The IDD project was started February 2000 with a duration of three years and involves both industrial and academic partners: Fiat CRF



(Torino), Magneti-Marelli SpA (Torino), PSA, Peugeot Citroen (Paris), Renault (Paris), DaimlerChrysler AG (Stuttgart), OCC'M Software GmbH (München), Università di Torino, Université de Paris Nord, XIII, and Technische Universität München.

ANALYSIS OF THE CURRENT PROCESS OF DESIGN AND GENERATION OF DIAGNOSTICS

The project started with an analysis of the current processes of each industrial partners with a focus on the integration of the diagnostic process and diagnosis-related processes into the whole design process of mechatronic subsystems.

Based on this analysis, a "merged process" has been developed that is based on the similarities recognized, ignoring details and small differences. The abstraction of this process is used as a comprehensive reference for the current design processes. This analysis and its consequences are presented in more detail in [Brignolo et. al. 01].

The core process the project is focused on is the "inner design loop" which is concerned with the design of the ECU-based control system and components. Each iteration involves the design of the control algorithms, failure-modes-and-effects analysis (FMEA), diagnostic development, implementation of the ECU (hardware and software) and verification of the algorithms, as shown in Figure 6. The verification step at the end of the first iterations is performed using models (software/ hardware in the loop), whereas, later, the physical system is used. Depending on the achieved results, there are several iterations, each one of them producing an advanced prototype.

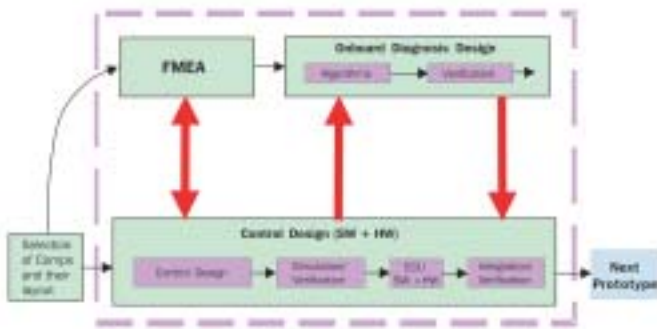


Figure 6: The reference process, one iteration of the inner design loop

A major deficiency of the current practice concerns the interaction between FMEA and the development of diagnostics on the one hand, and the development and design of control algorithms of the system on the other hand. Currently, these are carried out as two substantially separate tasks, despite the fact that there are important interdependencies.

As a consequence, requirements and constraints arising from one of these tasks can be dealt with by the other ones only in the next inner design loop, i.e. changes in the design of control algorithms can have impact on FMEA/ diagnosis only during the next inner design loop and vice versa, thus causing additional iterations and time delay.

TOOLS FOR A NEW PROCESS

Based on the analysis of the reference process and the required improvements, we propose a frame for a new process which is closely connected to a new tool architecture. In summary, the framework for a new process has to satisfy the requirement that the designers (the different experts involved in the design) should be supported in performing the different activities in an interleaved way and in evaluating different designs and in making choices about the best design of a system.

- Such a tight integration of different activities and the aim to perform them concurrently require the fast and reliable exchange of information about any changes in the design introduced by any of the activities. This is why we propose that the **model of the system being designed must play a central role in the new process.**
- The aim to update FMEA, diagnosability analysis and OBD generation quickly after a change and to consider different design alternatives in parallel establishes the requirement that these tasks can be effectively supported or automated by computer tools based on the model, i.e. they have to be **model-based tools.**

These tools rely on model-based systems and will be based on a common set of models and a common model-based diagnostic system core.

The new process and the respective tools should be integrated or combined with the simulation tools, that are currently used for the design of control strategies and typically based on quantitative models. In IDD, this is Matlab/Simulink. This requires software that transforms the models created in these environments into qualitative diagnostic models that form the basis for the model-based tools. The foundations of one of the implementations and a critical discussion of the practical experiences are presented in [Struss 02]. Figure 7 summarizes the overall architecture of the new design support system:

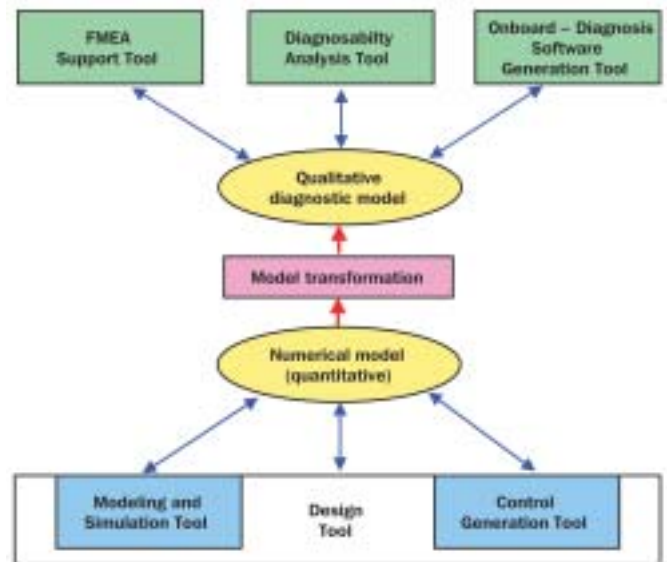


Figure 7: Tools architecture for the new process

STATUS AND FUTURE WORK

As of now, two different alternatives have been implemented to generate the qualitative diagnosis models from existing numerical models which both use Matlab itself to compute the tuples of the modeling relation. In addition, a library of qualitative models will be created manually that allows to configure the model based on the structural description only. Based on a use case analysis, the core of the diagnosability analysis tool and the model-based on-board diagnosis engine have been developed.

IDD will use a number of guiding applications with the goal to demonstrate how the diagnostic tasks described can be performed by using the new process and the new tools architecture. Furthermore, we aim to demonstrate how additional advantages of the new method can be achieved, e.g. optimization of sensor placement or deeper diagnostic performance. Thereby, the guiding applications serve, on the one hand, as case studies for the application of the new techniques and, on the other hand, as test cases and demonstrators of the results of the project.



The guiding applications chosen cover on the one hand different mechatronic systems with central ECU-functions, and on the other hand the general application of diagnostic tasks to multiplexed architecture systems. They include

- The air delivery system for diesel engines (Figure 8), comprising the exhaust gas turbocharging system and the exhaust gas recirculation system (EGR and the Common Rail Injection System (Fiat and Magneti-Marelli).

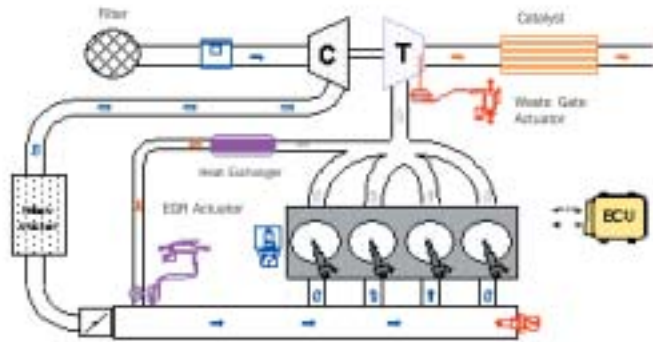


Figure 8: Guiding application: Air delivery system

- The cooling system (DaimlerChrysler AG), including an intercooler, which on the one hand increases the efficiency of the engine by cooling the compressed air and, hence, increasing the air charge rate, and on the other hand decreases NOx emissions by keeping the combustion at lower temperature (Figure 9).

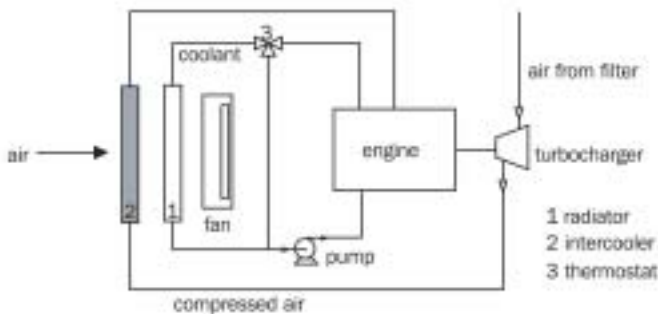


Figure 9 Guiding application: Cooling system

- The air conditioning system (Peugeot Citroën PSA) which consists of two loops that supply a cold heat exchanger and a hot heat exchanger (Figure 10).

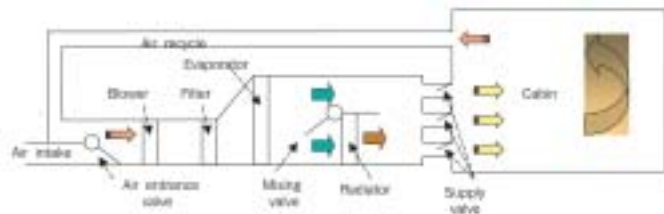


Figure 10 Guiding application: Air conditioning system

- The multiplexed architecture (Renault) involving ECUs, sensors, actuators, functions (EF = elementary functions), busses and data frames (Figure 11). The design engineer will be enabled to run a program directly on the representation of a designed architecture and receive the results of an analysis of the interdependency of faults and functions in this architecture.

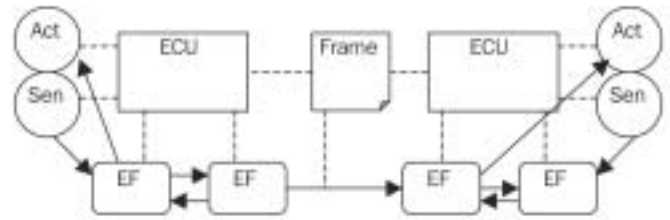


Figure 11 Guiding application: Multiplexed architecture

A first version of models for these guiding applications has been developed and will be used to validate and improve the model abstraction module and to evaluate the tools. By the end of the project in January 2003, we hope to demonstrate the utility of the tools and the benefits of the modified design process based on examples that are close to reality.

ACKNOWLEDGEMENTS

Many thanks to all partners in the VMDB and IDD projects, in particular to my direct collaborators Oskar Dressler, Alessandro Fraracci Martin Sachenbacher und reinhard Weber. The work reported here has been supported by the Commission of the European Union (Projects no. BE 95/2128 and no. G3RD – CT199-00058) and by the German Ministry of Education and Research (#01 IN 509 41).

REFERENCES

- [Bidian et al. 99] P. Bidian, M. Tatar, F. Cascio, D. Theseider-Dupré, M. Sachenbacher, R. Weber, C. Carlén: Powertrain Diagnostics: A Model-Based Approach, Proceedings of ERA Technology Vehicle Electronic, Systems Conference '99, Coventry, UK, 1999
- [Brignolo et al. 01] R. Brignolo, F. Cascio, L. Console, P. Dague, P. Dubois, O. Dressler, D. Millet, B. Rehfus, P. Struss. Integration of Design and Diagnosis into a Common Process. In: Electronic Systems for Vehicles, pp. 53-73. VDI Verlag, Duesseldorf, 2001
- [Bryant 92] R. Bryant: Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams ACM Computing Surveys, Vol. 24, No. September 1992
- [Cascio et al. 99] F. Cascio, L. Console, M. Guagliumi, M. Osella, A. Panati, S. Sottano, D. Theseider-Dupré: Strategies for on-board diagnostics of dynamic automotive systems using qualitative models, AI Communications, June 1999
- [de Kleer-Mackworth-Reiter 92] J. de Kleer, A. Mackworth und R. Reiter: Characterizing Diagnoses and Systems. Artificial Intelligence, 56, 1992
- [Dressler-Struss 96] O. Dressler und P. Struss: The Consistency-based Approach to Automated Diagnosis of Devices. In: Brewka, G. (ed.), Principles of Knowledge Representation, CSLI Publications, Stanford, pp. 267-311, 1996
- [Hamscher et al. 92] W. Hamscher, L. Console, J. de Kleer (eds.): Readings in Model-based Diagnosis, Morgan Kaufmann Publishers, San Mateo, CA, 1992
- [OBD 93] California's OBD-II regulation, section 1968.1, title 13, California code of regulation, Resolution 93-40, 1993
- [RAZ'R 02] Raz'r Version 1.6, Occ'm Software GmbH, see <http://www.occm.de>
- [Sachenbacher-Struss-Weber 00] M. Sachenbacher, P. Struss, R. Weber: Advances in Design and Implementation of OBD Functions for Diesel Injection Systems based on a Qualitative Approach to Diagnosis, SAE 2000 World Congress, Detroit, USA, 2000
- [Struss 02] P. Struss: Automated Abstraction of Numerical Simulations Models – Theory and Practical Experience. In: Sixteenth International Workshop on Qualitative Reasoning, Sitges, Catalonia, Spain, 2002



Whole Lifecycle Electrical Design Analysis

N.A. Snooke, C.J. Price

Department of Computer Science, The University of Wales, Aberystwyth, Ceredigion SY23 3DB.

D. Ellis

Ford Motor Company Limited, Room 15-3B/E-11, Dunton Research and Engineering Centre, Basildon, Essex SS15 6EE.

INTRODUCTION

Design analysis is conventionally performed late in the design lifecycle of a product once all the detailed design information is available. The drawback of this is that problems discovered at this late stage can be very expensive to fix. Qualitative reasoning technologies enable recently developed automated tools¹ to produce useful but less precise results at much earlier stages of the design. We discuss recent work to develop new tools that can combine the best of both worlds. Information can be acted upon as soon as a problem becomes apparent, without tedious and expensive repetition of analysis by experts. Early in the design, rough analysis can identify gross problems, whereas towards delivery time, detailed analysis can pinpoint complex problems that could not be identified precisely until enough detailed design decisions had been made.

The Dougal project², part of the UK Government's Foresight Vehicle Initiative, has developed design analysis systems that can give progressively better design analysis results as more detail is available about a vehicle's electrical design. This has been achieved through developing a range of simulation models that can be automatically constructed from schematic information. Results of the simulations are linked to a common notion of system functionality, which allows the results of the different simulations to be compared, and incremental changes to the results to be identified as the design evolves.

AUTOMATED DESIGN ANALYSIS USING QUALITATIVE MODELS

Qualitative models for many components are highly reusable and allow simulation to be performed very early in the lifecycle, perhaps six months to a year before detailed models are available. The qualitative results can easily be presented to users in the following ways:

Visualization of results: Simulation results can be used to colour the schematic within the CAD diagram, showing which parts of the circuit are active at any point in the simulation, so that the user can understand the effect of changing inputs. The direction of current flow is indicated by arrows. These visualization features are integrated with the other design analysis techniques, so that FMEA, for instance, can set up the circuit for simulation with specific faults induced on components, and visually demonstrate the effect of that failure on the circuit.

Abstraction of results via functions: The overall behaviour of a circuit such as a door-locking system can often be summarized by a single label such as locking or unlocking or locked. For a cruise control system, the overall behaviour might be summarized as accelerating, decelerating, cruising or deactivated. Most circuits will only need a few such functional labels. The software can determine the state of the overall system by examining the state of a small subset of the components in the circuit and labelling the functions that are occurring. The functional labels are used as the basis for written reports produced by the design analysis tools described in the next section.

The design analysis is based on the simulation, and can produce the following results:

What-if investigation: Once the schematic has been drawn, the engineer can alter inputs to the system, interactively flicking switches and activating sensors, and see the results of the simulation illustrated on the schematic. Many different scenarios can be tried – even potentially destructive ones – and answers found in a few hours to problems that might have otherwise taken weeks to solve. These investigations are the virtual equivalent of the physical breadboard models created by pegging out the system on a large board and tested against expected behaviour prior to the first product prototype.

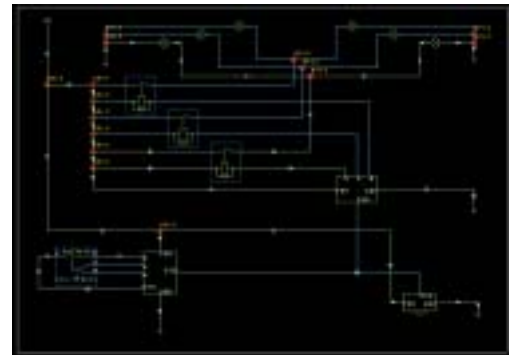


Figure 1: Extract from simulation

Failure mode and effects analysis: Failure mode and effects analysis considers the effect on an overall product of any (usually single) failure of part of the product. The software is able to simulate operation of the schematic when one or more components have failed. An FMEA report is produced which gives in English the effect of each possible failure of each component in a schematic on the functioning of the whole circuit. These are presented to the engineers in a standard FMEA report format. Table 1 shows an undoctored example row of an FMEA report produced by the software. The software has already detected a number of circuit faults early in development, and so saved a lot of money and cycle-time.

Assistance for FTA: Fault tree analysis highlights the combinations of failures that can affect the safety of a design. One of the uses of fault tree analysis is to compensate for the shortcomings of manual FMEA which typically only considers single failures. FTA is used to highlight all of the combinations of failures that will make a particular unwanted event occur. For example, such an event might be a vehicle's airbag firing when it should not. Alternatively, it might be to identify when the airbag will fail to fire. It is then possible to calculate an overall figure for how likely the unwanted event is to occur. Engineers calculate the dependencies in the fault tree by hand. The automated software can perform multiple failure FMEA. This provides all of the information that is needed to decide what combinations of failures can cause the unwanted event to occur. In addition, as vehicles become more complex, with ECUs programmed to mitigate the effects of known failures, it is likely to calculate the true effects of a combination of failures more accurately than an engineer mentally simulating circuit dependencies.

¹ See <http://www.firstearth.co.uk>

² For details see <http://www.aber.ac.uk/~dcswww/Research/mbsg/>



Sneak circuit analysis: In complex electrical systems, the interaction of several subsystems can cause further systems to be activated unexpectedly. A classic example is that of the cargo bay doors of a particular aircraft design, where operating the emergency switch for the cargo doors can cause the landing gear to lower unintentionally. Typically, such problems are caused when a wire, which was expected to provide current in one direction, is used in the opposite direction, causing a sneak path.

Sneak circuit analysis (SCA) is the process of identifying and eliminating such sneak paths where they might occur. We have implemented an automated sneak circuit tool capable of detecting classic sneaks [3]. The functions of the system will have already been declared for FMEA. It is necessary only to declare the combinations of inputs which should activate each function. Unexpected functions are then reported with the combination of inputs that caused the sneak. (See **Table 1** below.)

Unlike several other sneak circuit tools, it is not necessary to declare the direction in which current should flow through each wire. Neither does the algorithm produce spurious sneaks. However, it does have some drawbacks shared by other automated sneak circuit tools. It is necessary to simulate as a single unit all the subsystems that are suspected of interacting. In addition all combinations of inputs are tested ultimately placing restrictions on the number of inputs allowed.

Design verification: Given a formal description of the legal states in which a system can be, it is possible to analyze the operation of the design to ensure that the device cannot enter any illegal states. This is the least developed tool primarily due to the difficulty of obtaining a suitable system description. Systems are becoming more complex and so better specifications (e.g. state charts) are becoming a necessity and further development of this tool is likely.

ADVANTAGES OF EARLY AUTOMATED DESIGN ANALYSIS

Design analysis can be performed with very little effort early in the design lifecycle, and gross errors detected and rectified. This is the time when it is cheapest to fix problems, and so is a great improvement over performing analysis much later in the lifecycle.

Engineers can explore possible technical solutions without physically building many prototypes - that only becomes necessary once the majority of the problems have been ironed out and so one effect of these tools is to reduce the number of physical prototypes it is necessary to build.

The software simulates current flow through the circuit using state-based descriptions of complex components, and idealized resistors (with values of zero, load or infinity). This means that early modelling of components is simple and components are very reusable. The library of components needed is much smaller than is the case for numerical simulators.

It provides the best results possible before all information on specific components used is available.

DRAWBACKS OF EARLY AUTOMATED DESIGN ANALYSIS

The qualitative models can make it can be impossible to decide what will happen in a circuit. For example, if there is a short circuit, it is impossible to know whether a fuse will blow or wires melt unless the value of the fuse and the length and gauge of the wire are known. The early design analysis can only draw attention to a possible problem to be addressed when detailed design decisions are being made.

As extra information becomes available about the design, the engineers need to find other ways to check that problems raised by the early design analysis have been solved. The next section describes the different levels of information that become available, and how they are used to produce more precise versions of the results originally generated by the early design analysis.

IMPROVING ANALYSIS AS DESIGN INFORMATION INCREASES

Work has been undertaken to determine how the analysis results can be gradually improved and tracked as information becomes available during the design process. For electrical systems, there are three further kinds of extra information that might become available:

- Knowledge of power levels in the circuit
- Knowledge of numerical resistor values in the circuit
- Detailed numerical models for components in the circuit

KNOWLEDGE OF POWER LEVELS

The qualitative simulation described in the previous section uses three levels of resistance – zero, load and infinite. These are not enough to distinguish between levels of current. For example, a trickle current through a device might be used to provide a signal, where it is not enough to activate the device.

Some ambiguous situations can be resolved by adding further levels of resistance. We have implemented a scheme that allows an arbitrary number of levels. In practice, in present vehicles, a five level scheme gives some extra information in simulation. The levels are then: zero, low, medium, high and infinite. The presence of these distinctions allows the visualization to colour the circuit with the different levels of activity in the circuit. In a vehicle with a 12 volt battery, the visualization shows three levels of activity as green, yellow and orange. These three levels correspond to information level flow (for activating ECUs), activation level flow (for activating relays), and power level flow (for activating motors). The correct results can be obtained in many of the cases mentioned above, without any modelling compromises.

Item/Fn	Potential Failure Cause	Potential Failure Mode	Potential Failure Effect	Sev	Occ	Det
(23)	The component UNLOCK_RELAY has failure switch stuck at contact2.	For the first time, the "doors unlocking" function was achieved. Finally, regardless of any event change, the "doors locked" function was never achieved, and the "doors unlocked" function was always achieved.	Doors started unlocking unexpectedly. Doors unlocked unexpectedly. Doors failed to lock.	6	3	2
(24)	The component DEADLOCK_RELAY has failure coil blown.	When DRIVER_KEY_SWITCH was set to lock (3) the "doors locked" function was achieved unexpectedly. Also, when DRIVER_KEY_SWITCH was set to neutral (4) the "doors locked" function was achieved unexpectedly.	Doors locked unexpectedly.	6	2	4

Table 1: Example output produced by FMEA tool



KNOWLEDGE OF RESISTOR VALUES

Later in the design process, once decisions have been made about where to source components, precise values of resistors can be provided to the simulation, and the length and gauge of connections will be known. Once that is the case, most of the short circuit cases that were identified in early design analysis can be resolved.

DETAILED COMPONENT MODELS

For specific unresolved problems, or safety-critical systems, the engineers may choose to perform detailed numerical simulation using a tool such as PSPICE or SABER. The design analysis tools are interfaced to SABER, abstracting the detailed numerical results to produce the same English-level results that were provided by the qualitative simulator. This also provides a much more friendly interface to SABER for performing visualization work.

TRACKING ANALYSIS AS DESIGN DECISIONS ARE CHANGED

In the past, design analysis has usually been performed once during the development of a vehicle system. Where changes were made to the design after the analysis had been carried out, engineers would estimate the effects of the change, and limit the analysis to the perceived influence of the change.

Once the design analysis is automated, it is very little effort to repeat the analysis whenever a change is made to the design or when more detailed information becomes available. However, that is not the end of the problem. The analysis is only useful if engineers reconsider the results and take action on problems identified.

To minimize the engineer time (i.e. cost) for each analysis we have developed software to provide incremental FMEA results. When the automated FMEA is first performed, the engineer considers all results, and takes appropriate actions. When a change is made to the design, a new FMEA report is generated and the incremental FMEA software

compares the new results with the previous set of results. Results which have changed are presented to the user, along with any new results (for example, failures on components which did not previously exist). Typically, for a simple change, a very small number of results will change.

The use of functional labels allows incremental results between all the different types of simulation described in the previous section. The potential of this facility has not been completely explored, but very low cost, frequent analysis will minimize the detection time for any decision which caused a new design problem.

CONCLUSION

Automated design analysis based on qualitative simulation provides a very valuable tool for assisting engineers in dealing with the complexity of modern vehicle design and producing robust designs under shorter timescales. This work improves that facility in three important ways:

- The integration of more detailed types of simulation with the design analysis tools provides more accurate results as more information becomes available.
- The provision of analysis at several stages of design means that the engineer has the best possible analysis available at any stage of the design process. With the exception of the SABER analysis, this is made available with very little extra effort from the engineer.
- The incremental FMEA enables the tracking of all changes to the design and improved information as the design evolves with minimum effort from the engineers.

ACKNOWLEDGMENTS

The research described in this paper was supported by the UK Foresight Vehicle program under EPSRC grant GR/N06052, and done in collaboration with FirstEarth Limited.

MONET Project Office

Department of Computer Science
The University of Wales, Aberystwyth
Aberystwyth
Ceredigion
Wales SY23 3DB

Tel: +44 (0)1970 628521

Fax: +44 (0)1970 628536

Email: monet@aber.ac.uk

Webpage: <http://monet.aber.ac.uk>

