Integrating reliability analysis and diagnostics for complex technical systems

N A Snooke and **C J Price***
Department of Computer Science, University of Wales Aberystwyth, Ceredigion, UK

The manuscript was received on 25 April 2006 and was accepted after revision for publication on 20 February 2007.

DOI: 10.1243/1748006XJRR32

Abstract: Creating diagnostic information for complex technical systems has become a very costly and challenging process. Separation of the process of creating diagnostics from the processes of design and of reliability analysis (as is the default in many companies) means that reliability analysis work is often ignored or repeated by engineers creating diagnostics. The current paper explains the basic concepts behind the creation of diagnostic systems, and considers the extent to which reliability analysis results can be used to make the production of diagnostics more efficient. In particular, it considers how the automated production of reliability analyses can be fed into the process of creating diagnostic systems.

Keywords: fault diagnosis, fault prognosis, reliability analysis

1 INTRODUCTION

Human-designed systems are becoming ever more complex. As that happens, the issues of ensuring the safety of such systems, and troubleshooting failures of the systems, are becoming more important. The two issues are interlinked, as much of the work involved in performing safety analysis is reusable in the production of diagnostics and prognostics. The current paper uses the creation of automotive diagnostics as an example domain to illustrate both the concepts and the issues in using design information and the results of safety analysis for the production of diagnostics.

The lessons drawn out from the automotive domain are applicable to diagnosis of other human-designed physical systems, although not to naturally occurring systems such as the human body or the ecology of a planet. The reasons for this limitation are that design information and the mechanisms at work in the domain are available for the human-designed systems in a way that they are not for the naturally occurring systems.

**Corresponding author: Department of Computer Science, University of Wales Aberystwyth, Penglais, Aberystwyth, Ceredigion SY23 3DB, UK. email: cjp@aber.ac.uk*

2 CONCEPTS OF DIAGNOSTIC TROUBLESHOOTING

2.1 Basic primitives

1. *Line replaceable unit (LRU).* This is the lowest level component of a complex system that would be replaced when the system is deployed. It may itself be a complex subsystem, and may be taken back to a laboratory situation to be refurbished, but in the field, diagnosis does not go beyond replacing the LRU.
2. *Fault.* This is a problem with an LRU that may cause a higher-level system to misbehave. An LRU may have several potential faults that would cause different system misbehaviour. For example, an electrical relay might short out or stick open (with the consequence that a device would not be powered when it should be), or might stick closed (with the consequence that a device would be powered when it should not be).
3. *Failure.* For a complex system, a failure misbehaves at the system level as a result of a fault on one or more LRUs. For example, given the example fault of a shorted relay mentioned above, the failure might be that a flap on an aircraft fails to respond to the controls.

2.2 Stages of diagnosis

The whole of the diagnostic troubleshooting process can be described as made up of five tasks.

1. Problem identification.
2. Fault localization.
3. Fault identification.
4. Fault diagnosis.
5. Repair.

These tasks can be illustrated by considering an example of what they involve in the context of an automotive diagnostic system.

1. *Problem identification*. For a garage-based diagnostic system, problem identification is usually performed by the car driver (perhaps the driver brings the car to the garage complaining that the battery is flat every morning), or by some other system such as an on-board monitoring system (a warning light is illuminated indicating that the battery is not charging).
2. *Fault localization*. Given that the problem is with the battery charging/discharging system, then the fault can be restricted to the following causes:
   (a) problem with the battery charging system;
   (b) something causing a continual drain on the battery;
   (c) problem with the battery.
3. *Fault identification*. The three possible types of fault can be distinguished by applying a few simple tests to see whether the battery is charging and to see whether the battery holds charge.
4. *Fault diagnosis*. If the fault is in the battery charging system, then it can only have been caused by the run of wiring through the alternator to the battery. Probing with a voltage tester should allow identification of point of failure.
5. *Repair*. The repair process is fairly trivial in this case, either component replacement or repair of a faulty connection. In other cases, it may involve a more complex adjustment such as tuning the car.

Diagnostic systems for different types of complex system emphasize the five tasks to different degrees. For example, an on-board system for detecting problems with a helicopter engine would have a strong emphasis on the monitoring task. For that domain, detecting faults as they develop is important, whereas detecting the LRU responsible for causing the problem is not. On the other hand, a diagnostic system for a chemical processing plant is likely to be based around a monitoring system put into place for other purposes, and will have a strong emphasis on fault localization because the plant might consist of several miles of pipework, and this emphasis will significantly reduce the need for engineers to walk around the plant performing tests to achieve fault identification and fault diagnosis.

A related area is that of observability or testability or diagnosability. The value of all variables or the state of all components will not be easily available when monitoring or performing fault localization or fault diagnosis, and it is important for troubleshooting that it is possible to distinguish between problems that need different steps to solve them.

The issues of noise, uncertain data, and intermittent failures are also important for diagnosis, but will not be dealt with further in the present paper, as reliability information is of little help in addressing these issues.

2.3 Types of diagnostic system

Manufacturers of complex systems generate several different kinds of diagnostic software to support the deployment and operation of their systems.

1. *Off-line diagnostics*. This type of software originated in the operations manual of an organization, where there might be a table of possible failures, and for each failure would be listed the tests to be carried out, and the repair actions should a specific test identify the fault. This type of software is often implemented as some kind of rule-based or case-based system. Price [**1**] discusses when each of these choices are appropriate. He assumes that a problem (a failure or potential failure) has been detected through monitoring, and it is necessary to identify the fault(s) causing the problem.
2. *Monitoring systems*. Modern complex systems include many microprocessors, and so it is possible to monitor values as the system operates, and detect either values that are out of their operating range (e.g. engine temperature is too hot), or inconsistencies between values (e.g. a tank's level is decreasing although more liquid is detected flowing into the tank than out of the tank). There is a more indirect level of monitoring, where by-products of the functionality of a complex system are monitored in order to detect problems. For example, vibration monitoring on engines uses the existence of an unwanted effect to detect problems. There is much valuable research in the fault detection and diagnosis (FDD) community into indirect monitoring (e.g. reference [**2**]), but it will not be dealt with further in the current paper. The focus in this paper is on integration of design and safety analysis information into diagnostics and, for FDD systems, this integration comes after the monitoring has been carried out, not in order to accomplish it.
3. *Prognostics*. These are a logical extension of monitoring systems. For many complex systems, it

would be advantageous to detect a failure before it becomes catastrophic. A hot steel mill might jam because of a fault on one of the solenoid valves controlling the pressure on the mill. If a degradation in the performance of the valve can be detected, then it can be replaced before failure, avoiding shutting down the mill for several hours while the jammed steel is cut from the mill. For rotating machinery, FDD is often the most effective solution for prognostics, but there are other domains, such as the steel mill example, where design information can be used to direct the creation of prognostics.

4. *On-board diagnostics*. When a monitoring system detects a failure or predicts an incipient failure, it is possible to use further sensor information either to perform some kind of fault identification, or to store relevant information for later use in off-line diagnosis. On-board fault identification can be very important in safety-critical situations, in order to categorize the effects of a specific fault. For example, a problem on a vehicle might, depending on the type of fault, be such that the vehicle can 'limp home' safely, or might demand immediate cessation of operation.

3 TYPICAL SCENARIO FROM THE AUTOMOTIVE INDUSTRY

This section uses the automotive industry as an example domain to show the issues involved in building diagnostic software today. A very similar set of issues could be generated for the aeronautic industry or the process control industry.

3.1 Development process for present vehicle diagnostics

Most automotive manufacturers now produce several kinds of diagnostics. They generate on-board diagnostics for vehicle subsystems that are connected to an electronic control unit (ECU). The on-board diagnostics alert the driver to problems, and tell the driver when it is necessary to take the car for maintenance. They also localize the diagnostic problem and record diagnostic trouble codes (DTCs), indicating their localization conclusions. They produce service bay diagnostic systems. These can download DTCs from the diagnostic ECU on the vehicle, and use that information to perform fault identification and diagnosis from the DTC.

For electrical failures that are not covered by DTCs, and for many non-electrical systems, a procedure for localizing failures and pinpointing their cause is provided, either as part of the service bay diagnostic system, or as a physical manual.

The process of producing these diagnostics is very intensive in engineer time. It involves:

(a) decomposing the whole vehicle into manageable systems or subsystems;
(b) identifying LRUs in each system;
(c) exploring the consequences of each possible fault or combination of faults that could occur on an LRU;
(d) sorting the candidate faults by failure (so that when a specific failure is identified, all possible faults that could cause that failure can be identified);
(e) ordering the candidate faults for each failure so that diagnostic investigation will be executed efficiently;
(f) deciding and implementing a monitoring strategy for consequences in order to identify failures, and also to perform prognostics if possible;
(g) generating software to run in the service bay and to run on-board the vehicle.

When the candidate faults have all been identified, the correctness of their predicted failure effects are often verified by manually imposing each possible candidate fault on the physical LRU (e.g. breaking wires, shorting components to ground) and checking that the predictions are correct.

3.2 Business challenges for the diagnostic development process

The production of vehicle diagnostics in this manner is a growing challenge for automotive manufacturers for several reasons.

1. *The complexity of vehicle systems is increasing*. There has been a continuing trend over several decades towards greater complexity in vehicle electrical systems, because of greater numbers of features, because of the use of ECUs, and because of the pressures for greater vehicle efficiency and reduced emissions. That trend has greatly increased the complexity of the vehicle, and consequently of the associated diagnostics.
2. *Variants of vehicles demand different diagnostics*. On many vehicles, some features such as passenger air bags are optional. Other features, such as daytime running lights, are only mandatory in certain countries. Different variants of a vehicle may exhibit different failure behaviour for the same root fault, and demand different diagnostics. Producing different diagnostics for different variants can mean a great deal of extra work.
3. *Optimization of designs*. After a new vehicle design is produced, the design may change for several reasons. A generic problem with the design may have been found and fixed for future releases of

the vehicle, or sets of components in the design may be replaced with more economic components in order to reduce the cost of the vehicle. If the diagnostics have already been produced (which they should have), then under the present development process the diagnostics are unlikely to be modified to take account of such changes.
4. *Business organization*. The way in which the industry is structured often causes problems for the diagnostics work. The production of diagnostics is heavily software-based, and is usually done by a separate division from the product design group, or is even outsourced to another company. This increases the difficulty of producing good diagnostics, as much of the system understanding generated during design is unavailable for the team producing diagnostics.

The current paper addresses how design and diagnostic work can be structured to maximize the sharing of design and safety analysis information with the team constructing diagnostics, and how automated safety analysis work can help to address the other issues.

3.3 Technical challenges for the diagnostic development process

As systems become more complex, there are also technical challenges that gain greater significance.

1. *Single-fault assumption*. A common, although sometimes implicit, assumption of much diagnostic software is that a failure will be attributable to a single fault with the system. This assumption makes it much easier to generate the necessary diagnostics: the number of cases to cover is linearly related to the number of components in the system. If a diagnostic system is to cover failures caused by any combination of faults, then the number of possibilities grows exponentially as the complexity of the system grows. However, for complex systems, some of the most significant failures can occur because of multiple faults, and it is important that such failures are detected and dealt with.
2. *Tolerances*. For complex systems, a failure is not necessarily the result of a catastrophic fault on a component. It can be caused by a component gradually changing its tolerances, perhaps because of wear or rust. Worse, it could be because two interacting components change tolerances, where just one of them changing tolerance would not have caused the failure. 'No fault found' is a common diagnostic conclusion, and is often attributable to replacement or reseating of components that seem to be functioning correctly but have some tolerance problem.
3. *Intermittent failure*. One of the most frustrating things for a human diagnostician, and one of the most difficult challenges for diagnostic software, is when the same test done under the same conditions produces different results. Sometimes this can be a result of tolerance problems, but sometimes, especially where complex control software is involved, it can be because the problem only occurs in some obscure state of the system.
4. *Cost/benefit of test*. Where all possible information is available on-line, then a diagnostic system can work on perfect information. However, in many cases that is not true. It might be necessary to remove some trim from a vehicle (taking several minutes) in order to test one candidate fault, whereas another fault might be tested more easily. One test might rule out a number of possible faults, whereas another would only confirm or deny a single fault hypothesis. In some domains, the speed of diagnosis is an issue that must be taken into account. For example, if a methane production plant, was being monitored and one potential explanation for the observed symptoms was a large methane leak, it would be suggested that tests be performed to confirm or deny that possibility, even if it was not the most likely hypothesis. In general, fault identification and diagnosis involves a balance between many considerations in order to produce an effective and efficient diagnostic strategy.

4 INTEGRATING DIAGNOSTICS INTO THE PRODUCT LIFECYCLE

Many of the problems with the process of developing diagnostics described in the previous section can be lessened by making full use of the knowledge of system design and operation that was available to the original design team.

This section explores the advantages and costs of three ways of utilizing design information to produce diagnostic software.

1. Improvements through reuse of information: much information is produced during the design process that can be of use when building diagnostic software.
2. Automating the production of diagnostics: models of a system can be used to generate diagnostic software automatically.
3. Integrated use of information through the lifetime of a complex system: improvements to the design process coupled with technological advances could provide more effective diagnostics more efficiently.

4.1 Improvements through reuse of information

The design process produces several types of information that are of use when developing diagnostics for a system.

1. *The physical structure of the system*. The components that make up the system and the characteristics of each type of component are needed in order to understand how the system can fail, and the consequences of each possible fault.
2. *Specification of correct behaviour of the system*. This is not intended to imply formal correctness, an expression of the desired behaviour of the device, but rather the behaviour of the system as it has been designed. A formal specification often does not exist, but engineers are able to infer the expected behaviour of a system from knowing the structure and the correct behaviour of each component, if it is assumed that the design of the system is correct. Knowledge of the expected behaviour enables the detection of misbehaviour.
3. *Failure modes and effects analysis (FMEA) report* [**3**]. Where the designers have considered the consequences of each possible component fault, and calculated the system-level consequences of that fault, then the results help to provide a list of possible failures that the diagnostics need to cover, and the faults that could cause each failure. There are two problems when an FMEA report has been generated by engineers considering all possible faults on all components in the system. First, from studying genuine FMEA reports, it has been observed the engineers tend to assign different failure modes to the same failure, and the same failure mode to different failures. This misclassification can make it difficult to group failure modes for diagnosis. Such misclassification can be avoided by performing a FMECA (failure modes, effects, and criticality analysis). This forces the analysts to list together all of the faults that can cause a specific failure. The second problem is that hand-generated FMEA or FMECA is very person intensive, and usually only covers single faults, whereas some of the most challenging failures to diagnose are those caused by multiple faults.
4. *Fault tree analysis (FTA) report* [**4**]. Where a fault tree analysis (FTA) has been performed for a system, then it can give more detailed information about the occurrence of the FTA top event than is available from the typical FMEA report. Specifically, for that top event, it can provide all combinations of faults that can cause that top event to occur. Hurdle *et al.* [**5**] have done some interesting work on extracting diagnostics from an FTA report, but it is not a complete answer to the problem of producing good diagnostics, as FTA tends to be performed only for catastrophic top events.
5. *Component reliability information*. This information is helpful during fault diagnosis and repair. It can be used to focus diagnostic attention on the faults most likely to have occurred and to have caused the failure.

4.2 Automating the production of diagnostics

In response to the increasing overheads of producing diagnostics, vehicle manufacturers are beginning to automate the production of diagnostics. Simulation from a structural description of the system to be diagnosed, along with behavioural descriptions for components and knowledge of component faults, can be used to generate many of the low-level inputs to the diagnostic process [**6**, **7**]. In several domains, specifically electrical/electronic systems [**8**, **9**], and hydraulic systems [**10**, **11**], this technology is well demonstrated, and Struss and Price give details in reference [**7**] of how companies such as Daimler Chrysler, Ford, Scania, and Volkswagen are applying it to design and diagnosis.

For electrical or hydraulic systems within a vehicle, it is possible to provide structural information from a computer-aided design (CAD) tool as soon as a design is available, and generate simulations of the given system's behaviour. The structural information needed is the type of each component and the connectivity between the components. The type of the component is used to access a library of models of component behaviour. The behaviour of the overall system is generated from the behaviour of each component and the way in which the components are connected. The AutoSteve system for simulating electrical circuits is a good example of this technology [**9**, **12**]; it works for circuits containing very sophisticated electronic components with the electronic behaviour represented by state charts.

If the model of a correctly working component (such as a relay) is replaced in the simulation with a model of that component with a fault (e.g. a relay stuck open), then the behaviour of a system containing that faulty component can be simulated. If the behaviour of the correctly working system is compared with the behaviour of the faulty system, then the effect of the component fault can be identified. For example, in a lighting system with one of the relays stuck open, the effect of such a fault might be that the left headlight does not light when the headlight switch is turned on with the lights set to main beam. This is the system level failure for that fault.

The possible faults for every component in a system can be generated from the CAD description of the system and from knowledge of possible faults for each type of component. This provides a fault population. The effects of each possible fault can be

calculated using the procedure outlined in the previous paragraph. AutoSteve also abstracts the effects to the level of system function, in order to provide user-level descriptions of effects, and allows the efficient generation of multiple component faults. This process can be combined with reliability information to generate efficiently an FMEA report covering multiple faults [**13**]. Unlike many hand-generated FMEA reports, the failure results reported for each fault combination are consistent, and so are ideal for reusing in diagnostics.

In order to use this information for diagnosis, it needs to be reordered. All fault combinations that can cause a specific set of effects can be automatically grouped together. The specific set of effects (such as 'the left headlight does not light when it should') are then associated with every possible set of component faults that could cause that set of effects. When a symptom such as failure of left headlight is known, the results from this process can be used to identify possible component faults that could be responsible for it. Knowledge of component reliability can be employed to order the investigation of possible components during fault diagnosis, but needs to be combined with other considerations such as the cost of performing a specific diagnostic test in order to plan troubleshooting efficiently.

Workshop manual diagnostics, service bay diagnostics, and online diagnostics have all been produced in this way using automated FMEA output as a basis. The diagnostics are produced based on information provided by design engineers, and the effects of component failures will have been verified by reliability engineers. This provides a much closer link between design and diagnosis than is often the case in large corporations.

5 PRESENT RESEARCH AND FUTURE CHALLENGES

Looking to the future, vehicle manufacturers and their tier 1 suppliers are addressing increasingly serious challenges. Because the complexity and sophistication of vehicles is growing, it is becoming harder to predict interactions between vehicle systems, especially when failures occur. In response to that, many if not most manufacturers are at least investigating the kinds of automation of design analysis and diagnostics that have been outlined in the previous subsection. This is sometimes characterized as 'virtual prototyping': instead of finding out the drawbacks of a design on a very expensive physical prototype, exploration of the design is done on a virtual prototype.

However, many of the kinds of automated tool that have been deployed so far are essentially point tools. They can most profitably be used at a single point in the design process. Further research is aimed at integrating the use of design and reliability information throughout the lifetime of the product. An early example of this is given in reference [**14**], where safety analysis can be performed to different levels of accuracy as the details of an electrical design are made more precise. Changes to the reliability of the design can also be flagged as design changes are made.

The ideal end point of this activity would be the virtual vehicle: a model of the complete vehicle that can be developed and used throughout the lifetime of the vehicle. When a new vehicle is first conceived, then the requirements for the vehicle can be used to build a functional model of what the vehicle will be required to do. This might allow automatic specification of much of the complex equipment in the vehicle. As the design is refined by the engineers, then the extra information should be incorporated into the model of the vehicle from databases of component models.

As the design process progresses, it will be possible to perform automated tasks such as verification that the design meets the requirements, FMEA, system simulation, diagnosability analysis, production of diagnostics, and generation of control software. As variants of the new vehicle design are produced, all this work can be repeated with much less effort, reusing all information that is shared with the original model. When the vehicle is finally disposed of, the virtual vehicle can be used to plan disassembly and efficient disposal of materials.

If the use of design information is to be automated across the lifecycle of the vehicle, and for the variety of purposes described above, then there are a number of challenges that need to be addressed.

1. *Integration of different types of model.* One issue that is growing in importance is the ability to reason as effectively as possible about a system where different subsystems are specified with different degrees of detail: perhaps only a qualitative model exists for one subsystem, a functional model for several others, while one or two subsystems can provide detailed numerical models. Combining these different levels of information is not possible at present, but will become vital if the virtual vehicle is to be realized.
2. *Emphasis on software.* The modelling of the action and influence of software is an issue for almost any advanced man-made device or system. For example, in the automotive domain, ECUs containing many thousand of lines of software control the state of vehicle systems, and often perform monitoring, diagnosis, and reconfiguration of systems. It is necessary to incorporate the actions performed by software when modelling

the behaviour of the overall system, in order to understand the state of the device, and perform device-specific analysis.

3. *Integration of effects that cannot be efficiently modelled.* Even within fault diagnosis, there are limitations where problems are caused by faults outside the domains modelled. A simple example of this is provided by the in-car entertainment system. One reason why the radio might not work would be that the aerial was no longer connected to the radio. This would not be modelled as part of the electrical system of the car, and so would not be suggested as a cause of the radio failing. In theory, it would be possible to model all aspects of a vehicle and generate diagnostics for all aspects of the vehicle from the models. In practice, it is significantly inefficient in examples such as the aerial in the radio explicitly to model the effect accurately, and so it needs to be possible to model such effects just as functional dependencies.

6 CONCLUSIONS

Producing diagnostics for complex systems has become a very costly and time-consuming business, and the current paper has outlined both the issues involved in producing diagnostics, and the ways in which the cost can be reduced. Much of the work involved in designing a system and analysing its safety and reliability is reusable in the production of diagnostics, and maximum use of that information can reduce costs significantly.

In the automotive industry, automated generation of both FMEA reports and of diagnostics based on the structure of the design and the behaviour of the components is becoming commonplace at the system level, at least for domains such as electrical and hydraulic systems. The techniques described here have also been deployed in other domains where there is a complex system to be analysed and diagnosed, and sufficient design information is available.

REFERENCES

1 **Price, C.** *Computer based diagnostic systems*, 1999 (Springer Verlag, New York).

2 **Patton, R., Frank, P.,** and **Clark, R.** (eds.) *Fault diagnosis in dynamic systems: theory and applications*, 1989 (Prentice Hall, New York).

3 **Mcdermott, R., Mikulak, R.,** and **Beauregard, M.** *The basics of FMEA*, 1996 (Productivity Press, New York).

4 **Andrews, J.** and **Moss, B.** *Reliability and risk assessment*, 2nd edition, 2002 (Professional Engineering Publishing, Bury St Edmunds and London).

5 **Hurdle, E. E., Bartlett, L. M.,** and **Andrews, J. D.** System fault diagnostics using fault tree analysis. Proceedings of the 16th Conference on *Advances in reliability technology symposium (ARTS)*, Loughborough, Leics, UK, 2005, pp. 203–222.

6 **Picardi, C., Bray, R., Cascio, F., Console, L., Dague, P., Dressler, O., Millet, D., Rehfus, B., Struss, P.,** and **Vallee, C.** IDD: Integrating diagnosis in the design of automotive systems. In Proceedings of the *ECAI-02*, 2002 (IOS Press, Lyon).

7 **Struss, P.** and **Price, C.** Model-based systems in the automotive industry. *AI Magazine. Special issue on Qualitative Reasoning*, 2003, **24**(4), 17–34.

8 **Price, C.** Function directed electrical design analysis. *Artif. Intell. Engng*, 1998, **12**(4), 445–456.

9 **Snooke, N.** Simulating electrical devices with complex behavior. *AI Communs*, 1999, **12**(1,2), 45–59.

10 **Sachenbacher, M.** and **Struss, P.** Fault isolation in the hydraulic circuit of an ABS: a real-world reference problem for diagnosis. Working notes of the 8th International Workshop on the *Principles of diagnosis*, (DX-97), Mont-Saint-Michel, France, 1997, pp. 113–119.

11 **Hawkins, P.** and **Woollons, D.** Failure modes and effects analysis of complex engineering systems using functional models. *Artif. Intell. Engng*, 1998, **12**(4), 375–397.

12 **Price, C.** AutoSteve: automated electrical design analysis. In Proceedings of the *ECAI-2000*, 2000, pp. 721–725 (IOS Press, Berlin).

13 **Price, C.** and **Taylor, N.** Automated multiple failure FMEA. *Reliability Engng System Safety J.*, 2002, **76**(1), 1–10.

14 **Price, C., Snooke, N.,** and **Lewis, S.** A layered approach to electrical safety analysis in automotive environments. *Computers Ind.*, 2006, **57**, 451–461.