Aberystwyth University

A layered approach to automated electrical safety analysis in automotive environments

Price, Chris; Snooke, Neal; Lewis, Stuart

Published in:
Computers in Industry

DOI:
10.1016/j.compind.2006.02.001

Publication date:
2006

Citation for published version (APA):
Price, C., Snooke, N., & Lewis, S. (2006). A layered approach to automated electrical safety analysis in automotive environments. *Computers in Industry*, *57*(5), 451-461. https://doi.org/10.1016/j.compind.2006.02.001

A layered approach to automated electrical safety analysis in automotive environments

C.J. Price *, N.A. Snooke, S.D. Lewis University of Wales, Aberystwyth, UK

* Corresponding author at: Department of Computer Science, University of Wales, Aberystwyth SY23 3DB, UK.
Tel.: +44 1970 622424; fax: +44 1970 628536. E-mail address: cjp@aber.ac.uk (C.J. Price).

Abstract

Software support for the automotive electrical design process is vital, as many of the safety analysis tasks needing to be carried out, while complex, are repetitive and time consuming. Such support is required throughout the design process, but the available commercial tools are only appropriate at specific points in the design process—providing either an early rough analysis or a late but detailed analysis. This paper describes how the capability and utility of safety analysis software can be improved through separating the types of knowledge used into layers. This allows the maximum amount of information to be reused as the design evolves, and enables software tools to track the consequences of changes to the design so that the repercussions of any design change can be understood. The software capability described has profound implications for the design process. Previously, engineers performed a snapshot design safety analysis at some point in the design process, even if they had an automated design safety analysis tool to assist them. The process and tool arrangement described in this paper enables engineers to continually monitor the status of a design, noting the implications of any changes or refinements to the design.

Keywords: Safety analysis; Model-based reasoning; FMEA; Diagnosis; Automotive

1. Introduction

The complexity of modern automotive designs, especially in the electrical/electronic domain, has gradually increased over the past 50 years. Where analysis of the safety aspects of the design, through disciplines such as failure modes and effects analysis (FMEA) [1] is performed without computerized assistance, it is usually carried out very late in the design process, because it is so time consuming. Design safety analysis software is needed to automate this process and to identify problems early on, when it is comparatively cheap and easy to solve them.

Design safety analysis tools can be built that use the results of numerical simulators such as RODON [2] or SABER [3], but the detailed level of information needed to employ such simulators is not available until near to the end of the design process, when component suppliers have been identified, and precise details such as the length of wires have been established. In addition, the detailed numerical results provided by such simulators are not appropriate for reporting design safety problems automatically.

Design safety analysis tools based on qualitative reasoning [4] are an effective way of obtaining early feedback on potential problems with designs, and can typically be employed as soon as the functionality and structure of the system is known. However, they become less effective later in the design process. As further design details become available, such tools are unable to use the extra information to resolve some of the ambiguities of operation that were present in the design before detailed decisions had been made. Between them, the qualitative reasoners and the numerical simulators offer a number of point tools providing either valuable early feedback or precise late analysis [4,5].

This paper describes how the capability and utility of design safety analysis tools can be improved through separating the types of knowledge used into layers so that the maximum amount of information can be reused as the design changes, and through providing tools that track the consequences of changes to the design so that the repercussions of any design change can be understood.

A case study is presented of an electrical lighting circuit from a modern vehicle that evolves several times during its design development, showing how the analysis results change. Computer-based support for managing the changing design is described. While much of the content of this case study concentrates on the design of electrical systems with electronic components, the principle of modeling the system in layers for reuse is not limited to that domain, and applicability to other domains is considered in a later section.

Separation of the modeling knowledge for reusability throughout the design process has profound potential implications. Previously, engineers performed a snapshot design safety analysis at some point in the design process, even if they had an automated design safety analysis tool to assist them. Structuring modeling knowledge so that it can be used with different simulators as the precision of the design increases enables engineers to continually monitor the status of a design, noting the implications of any changes or refinements to the design.

The paper is structured in the following way:

- Section 2 gives an overview of the types of design safety analysis that can be automated based on a simulation of the underlying system.
- Section 3 explains briefly how qualitative reasoning about electrical systems can be integrated into a simulator capable of providing the information needed to perform design safety analysis, and shows the drawbacks of a purely qualitative design safety analysis.
- Section 4 shows how replacing the lower layers of the simulator allows it to provide more accurate results as a more detailed design becomes available.
- Section 5 presents a case study that illustrates the improved results that become available as more detailed design is done.
- Section 6 considers the implications of continuous design safety analysis on the design process, and shows how it can be employed to track the implications of changes to a vehicle design.
- Section 7 considers related work and the application of these techniques outside the domain of automotive electrical systems.

2. The automation of design safety analysis

Automated assistance for a range of design analyses can be provided if a simulation of a system can be performed that provides information about the system states under which system functions are achieved. This section gives a summary of some useful safety analysis techniques and how they can be automated.

2.1. Virtual prototyping

In a traditional design process, engineers will peg out an electrical design once the design is finalized, in order to test that the design works correctly. The engineers will have a set of tests that need to be carried out (e.g. turn the ignition on and turn this switch, the sidelights should come on). They will work through the tests and check that all results are as expected. This process can be performed virtually on a schematic of the system if a simulation is available. The ability to identify possible inputs (switches and sensors) and to specify results at the level of function (e.g. the sidelights are on), makes it very easy to specify tests and results [4].

2.2. Failure modes and effects analysis (FMEA)

Failure modes and effects analysis (FMEA) [1] is a design discipline used extensively in the aerospace and automotive industry. Every possible failure that can occur for a specific design is considered, and the effects of each failure on the operation of the overall system are calculated, in order to identify severe, frequently occurring failures, and eradicate them from the design if possible. FMEA can be performed hierarchically, with the failure modes at the lower level producing effects at the higher level. When FMEA is performed at the system level, the failure modes are component failures, and the effects are loss of system functionality or unexpected activation of functionality, due to the component failure(s).

System design FMEA can be automated using the results of repeated simulations, as long as the simulation is component-based, and descriptions of possible component failure behaviors are available [6].

Initially, the system being analyzed is simulated with all components in working order. The functions that occur in each state of the system are noted. Then, for each possible component failure (or combination of component failures), the simulation is re-run with faulty components. The functions that occur in each state for the faulty versions of the system are noted. The effect of a failure is the change in functionality observed. For example, if in the working version of the system, the sidelights and headlights come on when switch 3 is on, but in the version of the system with wire 23 broken, only the sidelights come on, then the effect of wire 23 being broken is that the headlights do not come on as intended. Assessments of severity and detectability can be associated with function, and assessment of occurrence frequency can be extracted from component failure information, and so each failure can be tagged with its significance automatically.

The automation described performs the arduous part of the analysis, but the engineer must then consider the most serious failures, and decide whether any action can be taken to avoid their unwanted effects. Once that has been done, the FMEA should be performed again, in order to see whether there are any unexpected new effects caused by the change to the system design.

2.3. Sneak circuit analysis

A sneak path is a route through a system by which current can flow causing unintended activity in the system. A simple but classic occurrence of a sneak path in a real design is documented by Savakoor et al. [7], where the interaction between an airplane’s landing gear and the switching mechanism relating to the cargo door can cause the landing gear to be inadvertently deployed while the airplane is in flight, because current can take an unexpected path through the circuitry.

The increasing complexity of modern vehicle circuitry means that the potential for sneak paths is increasing, with consequences for vehicle safety and reliability. Analysis to identify sneak paths is necessary to reduce the occurrence of such problems.

Simulation can be used to automate sneak circuit analysis [8] if a declaration of the states under which functionality should occur is given (e.g. the sidelights should be on when either switch 3 or switch 4 is on). All operating states can be automatically explored (by changing all inputs to the system—switches, sensors and internal states of electronic components), and sneak circuits can be recognized as unexpected activation of function (e.g. the sidelights come on when switch 6 and switch 7 are on, even though neither switch 3 nor switch 4 is on).

2.4. Functional design verification

The simulation of the system can be explored to produce all possible states that the system can reach, and a compact state chart can be produced that shows all states that the system can possibly reach. The design engineers can use this to identify any unexpected relationships between the attainable states [9]. This is a generalization of the sneak circuit analysis work, combined with an effective algorithm for compressing state charts, and can provide important information about the system's operation not available elsewhere. The engineer can use the generated state chart to verify that the system does not have any unexpected behavior.

2.5. Workshop and on-board diagnosis

When a new vehicle is designed, it is necessary to produce both on-board diagnostic systems and workshop manuals for the vehicle. So, for example, when designing the central door-locking system for a vehicle, the manufacturer will produce on-board diagnostic software that will run in the vehicle. This software will both detect failure of the system (perhaps a door failed to lock as expected), and indicate to some degree the reason why the failure occurred (for example, because the lock signal was not received by the electronic control unit). The on-board diagnostic software might indicate the failure and instruct the driver to take the vehicle to the garage. The workshop manual (or, in some cases, an online service bay diagnostic assistant) would then show the technician what to investigate to efficiently solve the customer's problem.

The theory of model-based diagnosis, generating possible explanations for symptoms from a knowledge of system structure, has been developed over several decades, both for single faults [10] and for multiple faults [11,12]. More recently, simulation-based generation of diagnostics has been explored in the automotive domain for the production of on-board diagnostics, of workshop manuals and of computer-based service bay diagnostics [4,13–15]. Essentially, a set of links between symptoms and the underlying failures that might cause them are generated, and arranged using knowledge of the domain in such a way that it is possible to perform efficient diagnosis.

Where models are being used for failure modes and effects analysis, there is a strong link between generating diagnostics and the work done already for FMEA. For FMEA, analysis begins with each potential failure of each component, and generates the possible effects of the failure (or symptoms, in diagnostic terms). Many different failures may have the exact same effect. For diagnosis, the effects or symptoms are the starting point. Typically, the driver of a vehicle will detect a problem – a lamp fails to illuminate, or the engine is slow to respond to the throttle – and will want to know what component failure(s) caused the observed behavior, in order to repair the system. Because of the consistency of the automatically generated FMEA reports (unlike those produced by engineers as free text reports), simulation-based results that were generated for FMEA can be rearranged to produce a list of all component failures that could cause each specific effect. This is discussed in more detail in [16].

2.6. Process implications of automated design safety analysis

When performed by an engineer without tool support, all of the design safety analysis disciplines outlined in this section are extremely repetitive and time consuming to perform. Merely performing them once on each system in a vehicle is a great deal of effort, and so they tend to be carried out very late in the design process when changes to the design are less likely—if they were performed earlier, then changes to the design would necessitate repeating the analysis. Late performance of analysis is not ideal, as changes to the design at that point will be much more costly, and have repercussions outside of merely changing the design. For example, if a problem with a system is found during field trials of a prototype, then the diagnostics may well have been produced, and a change to the system may mean changes to the workshop manuals, because the diagnostics are dependent on the structure of the system.

Implementation of automated design safety analysis tools based on qualitative reasoning provides the ability to perform design safety analysis with very little effort early in the design process, so gross errors can be 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 process. 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 [17].

A further process improvement that has been experienced in practice is due to the sharing of models and results between design engineers and diagnostic system builders. Many companies experience discontinuity between the two activities, where the design engineers produce a design and then pass it as a drawing to the diagnostic system builders, who need to identify what might happen if there is a failure of the working system. Now, the design engineers and diagnostic system builders share models that can be used to simulate the system. An initial set of diagnostics can also be automatically generated from those models [4].

The next section will explore how qualitative reasoning can be structured in order to generate the design analyses described in this section in an efficient manner, and will explain why it is not the complete solution to what is needed to integrate automated design safety analysis throughout the design process.

3. Architecture of automated electrical design safety analysis

3.1. Structure of the architecture

Three layers of reasoning are employed in order to perform qualitative simulation for electrical design safety analysis. These three layers separate out the different facets of the system in order to facilitate ease and reuse of model building:

- a layer of reasoning about function;
- a layer of reasoning about a component’s behavior;
- an electrical qualitative grid reasoner.

Fig. 1 depicts an example of the relationship between the three reasoning layers, showing for a very simple circuit containing a switch, a lamp, a relay, some wires and a battery how an electrical network is generated from the schematic, and that the system level functions are abstracted from the state of significant components.

The top layer provides teleological information [18]. It describes the intended functionality of the system being simulated. This information is independent of how the system is implemented and so is very reusable from one design to the next. For example, a windscreen wash system might be implemented as an electrically driven system or a hand pumped system. The function is the same, that water is squirted onto the windscreen. Similarly, a door-locking system might be implemented with a complex set of relays, or with a computer controlling the locking, but the functions implemented could be identical. The functional information for a system can be defined for a typical vehicle, and reused as the system is redesigned, with modifications as new functionality is added to a vehicle.

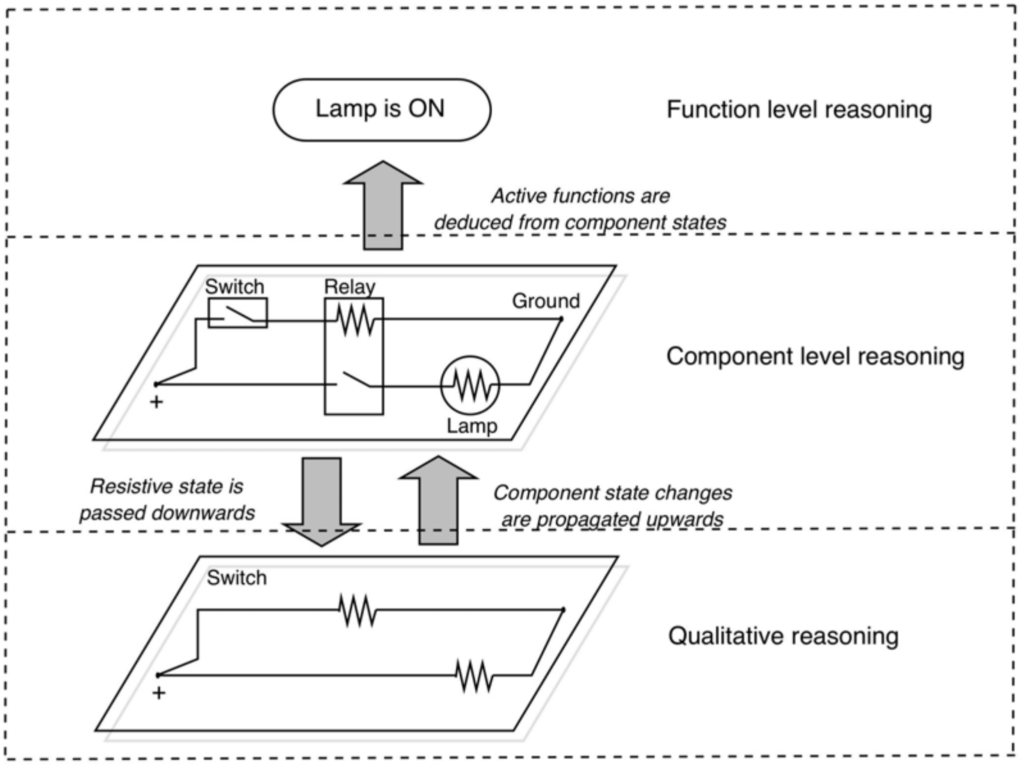

Lamp is ON
Function level reasoning
Active functions are deduced from component states
Switch
Relay
Ground
Component level reasoning
+
Lamp
Resistive state is passed downwards
Component state changes are propagated upwards
Switch
Qualitative reasoning
+

Fig. 1. Levels of electrical simulation

The middle layer describes components at the level at which the engineer defines them. It coincides with the components that an engineer would draw on a schematic, or would select for purchase from a supplier. Defining behavior at the component level means that components can also be defined in a reusable manner—the behavior of a relay can be defined, and whenever an engineer adds a relay component, the simulator will be able to use the same defined behavior. While components are reusable, the major representation task is at the component level. Components are assigned an appropriate network structure and a behavioral description. The physical network structure contains dependent resistors that can take values of zero, load or infinity, depending on the state of the component. The state of a component is changed by outside influence (turning switches and sensors on or off), or by responding to changes in the current through specified resistors within the component in the present state of the system. For example, the relay's state is changed when current flows through the coil of the relay: the value of the switch resistor in the relay is then changed from infinity to zero, reflecting that the switch inside the relay will have closed in response to the current flowing through the coil.

The lowest layer deals with underlying primitives of the electrical domain—resistors and connections. It is needed in order to determine what events occur next for an electrical system in a given state. This level of representation is generated from component-level models, as will be described in the next section.

3.2. Simulation with the architecture

The lowest layer of the architecture is the electrical qualitative grid reasoner. In Section 4 of this paper, this reasoner will be replaced by simulators with different levels of precision, but this section concentrates solely on a qualitative reasoner capable of deciding where current is flowing in a network. It takes a representation of an electrical system as a network of qualitative resistors connected to battery and ground, performs analysis, and establishes which parts of the system are active, using the method described by Lee [19].

Use of the electrical qualitative grid reasoner is driven by the middle layer of component-level reasoning. The middle layer generates a qualitative grid (or network) of resistors from component models, using information from the electrical schematic describing the components and how they are linked, and taking into account the state of each component. Changes in the electrical qualitative grid layer can change the state of the components (e.g. if current flows through the coil of a relay, then the relay will close). The change of state of the relay will change the configuration of the electrical grid in the qualitative layer (as current may now be flowing through the switch of the relay). The changes may continue to propagate through the circuit (e.g. the current flowing through the switch of the relay powers a lamp filament, and so the lamp is lit). A detailed example of this process is given in [20].

The component-level simulation consists of a sequence of dc electrical analysis steps controlled by the ability of the component-level models to change the value of the qualitative resistor values. Ambiguities in the events sequence are resolved by a simple model of time that considers all component state changes within a qualitative timeslot as concurrent, i.e. actions do not take effect until all events in a given timeslot have been completed.

Information about the state of each component at each step during a simulation is too detailed a set of results for use in safety analysis, and so the detailed behavior of the circuit needs to be abstracted to obtain a teleological description of the overall behavior of the system in terms appropriate to the engineers. This is done by identifying the states of significant components, and using them to determine the functionality of the overall system [21]. Typically, this will be focused on the effectors of the system (motors, lamps, controllers).

3.3. Advantages and drawbacks of qualitative reasoning about circuits

The advantages of using qualitative reasoning as the lowest layer of the three layered architecture described are:

- Early modeling of components is simple and components are very reusable. The library of components needed is much smaller than is the case for numerical simulators.
- Qualitative reasoning provides the best results that are possible before detailed information on the specific technical specifications of the components used is available.

However, there are also drawbacks to purely qualitative analysis of circuits:

- Because only idealized resistors are used, 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 safety analysis can only draw attention to a possible problem to be addressed later, when detailed design decisions are being made. _
- Some types of design safety analysis cannot be addressed with purely qualitative models. For example, quantitative information is needed to identify whether fusing is correct so that maximum loads do not cause fuses to blow when there is no fault.
- As extra information becomes available about the design, the engineers need to find other ways to verify that problems raised by the early design safety analysis have been solved. For example, this might mean using the SPICE simulator to get detailed results for a specific failure case [22].

Qualitative reasoning enables the detection of potential problems early in the design process with comparatively little effort. As the design process progresses, more detailed information about components becomes available for analysis, but a system purely based on qualitative reasoning is unable to use it. The next section considers the types of more detailed information that become available, and shows how they can be used to produce more precise versions of the results originally generated by the early design safety analysis.

4. More accurate model-based reasoning

The analysis results should be gradually improved and tracked as extra information becomes available during the design process. This section shows how this can be done in the electrical domain. The qualitative analysis described in the previous section can be carried out once the first schematic has been designed, and the functional behavior of the components is defined. As the design progresses, components are sourced from different manufacturers, and the logical design is enhanced with physical information such as the physical size of wires, and how they are routed around the vehicle. As this happens, there are three kinds of extra information that will become available:

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

4.1. Knowledge of resistor levels

The qualitative reasoning described in the previous section uses three magnitudes of resistance—zero, load and infinite. This granularity is not enough to distinguish between different levels of current. For example, a trickle current through a device, say a motor, might be used to provide an information signal, without being enough to activate the device. The qualitative reasoning cannot distinguish between the two levels of current, and so tests to decide whether current levels are high enough for activation of the device must be left to later in the design process.

Some ambiguous situations can be resolved by adding further levels of resistance. Lee et al. [23] have developed a scheme that allows an arbitrary number of levels. In practice, in present vehicles, a five level qualitative scheme provides the most useful extra information in simulation. The qualitative resistance levels are then: zero, low, medium, high and infinite. The engineer is often able to specify these levels quite early in the design process.

The presence of these distinctions allows visualization to illustrate the circuit with the different levels of activity in the circuit. In a vehicle with a 12 V battery, visualization can show three levels of activity as orange, yellow and green. These three levels correspond to information level flow (e.g. for activating ECUs), activation level flow (e.g. for activating relays), and power level flow (e.g. for activating motors). Allowing multiple values for only the resistance variables avoids the well-known problems associated with many variables containing multiple landmarks [24] and provides a similar number of levels of current as the output of the analysis. The levels must have an order of magnitude relation. In practice, the resistance of an arbitrary number of resistors connected in series cannot be greater than the qualitative resistance of the largest.

In the example given earlier in this section, where a trickle current through a motor provides a signal but does not activate the motor, the abstraction to system level functionality can be refined to recognize that only a large current will power the motor, and the trickle current will not then be mistakenly expected to power the motor.

This scheme can be implemented simply by replacing the lowest of the three reasoning layers described in the previous section. The qualitative reasoner is replaced by the multi-level qualitative reasoner, and information about resistor levels is added to each component type. Extra information needs to be added to the component model structure, replacing the load values of resistors with high, medium and low loads. This is fairly obvious, depending on the function of the resistor. The simulation then works as before, with functions being abstracted from system state, and the design safety analysis results are provided in the same format.

4.2. Knowledge of resistor values

Later in the design process, once design decisions have been made about specific components to be used, and physical decisions have been made about where to route wires, then precise values of resistors can be provided to the simulation, and the length and gauge of connectors will be known. Resistance values for wires can be automatically calculated from length and gauge details, and quantitative values for resistors in other component models can be easily found by testing the components. Once that information is available, most of the short circuit cases that were identified in early design safety analysis can be resolved, as is shown in the case study in a later section of this paper. Before numerical resistor values were available, it was impossible to calculate whether a fuse would blow or a wire melt (if the fusing was wrong). Once resistor values are known, these ambiguous cases can be resolved.

This scheme can also be implemented within the three layered framework outlined earlier. The lowest of the three reasoning layers described in the previous section is again replaced, and quantitative results are mapped onto qualitative values in the component model. The qualitative reasoner is replaced by a numeric simulator, PSpice [22], and the network simulation is done in PSpice. The resultant values are mapped onto the qualitative values in the state-based component models at the component level, and the state of components altered in response in the same way as for the qualitative models.

Using this approach, only two additional pieces of information are required from the engineer. Firstly the numerical resistance values for components, and secondly the thresholds used to map quantitative results (current flow) into the qualitative ones understood by the component-level behavior models. The ability to define the range of numerical values for each qualitative range at the component level is useful since a ‘‘negligible’’ current level (qualitative zero) may be different for different components. For example, the current needed to activate a component might be at least 20 mA for a certain type of relay but 800 mA for a large motor. A range for negligible values of current is also necessary using most numerical simulators because the qualitatively useful values zero and infinite cause problems for the numerical network solvers.

4.3. Detailed component models

For specific unresolved problems, or safety-critical systems, the engineers may choose to perform detailed numerical simulation using a commercially available tool such as SABER [3] or PSpice [22], with complex numerical component models. We have integrated SABER with the existing design safety analysis tools. This works by quantizing the detailed numerical results given by SABER and producing the same functional descriptions of results that were provided by the qualitative reasoner. As well as producing the type of design safety analysis results only previously available from the qualitative reasoning, this work also provides a much more friendly interface to SABER for performing visualization work.

In this case, both the lower and middle of the three reasoning layers have been replaced by the use of a numerical simulator with detailed numerical models. These models are time consuming to build for components, and are much less robust when simulating failure situations (as happens in the automated FMEA) than the qualitative models. Experience of deploying these tools in industry indicates that it is impractical to perform analysis for all systems in a vehicle in this way. However, where detailed simulation is needed in order to explore the design implications of factors such as motor inrush currents or lamp filament temperatures, then such detailed modeling may be appropriate.

4.4. Analysis results

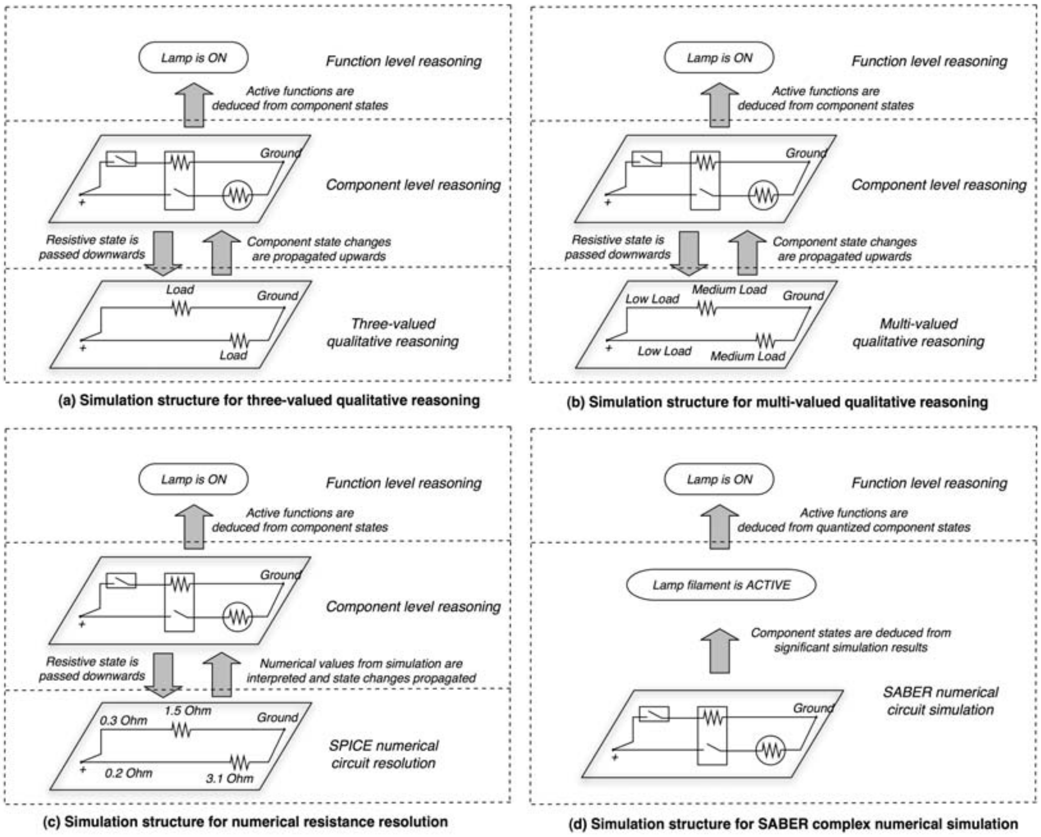
Fig. 2 depicts the interaction between the layers for the four different granularities of simulation about circuit behavior that have been described.

Fig. 2(a) depicts the original three layered architecture described in Section 3, where qualitative reasoning about resistors is used to calculate what is happening at the component level, and that is abstracted to the functional level where analysis reports are generated from facts such as a lamp is on when a circuit is in a specific state.

Fig. 2(b) depicts the architecture for making use of resistor level information. The simple qualitative reasoner is replaced with one capable of making distinctions between different types of resistor, but otherwise the layers are unchanged.

Fig. 2(c) depicts the architecture that can be employed once components have been sourced with specific manufacturers. Resistances across components and along wires can be measured, and the values entered into the schematic models. The lowest layer can then be replaced with a numeric simulator calculating current throughout the resistive network, with the results being propagated to the component level, where they affect the state of the components. The results from the component level are still abstracted to the functional level in the same way.

Fig. 2(d) depicts the case where complex component models are specified within the numeric simulator. This replaces the lower two layers of the architecture with the numeric simulator. The numeric simulator is responsible for simulating the different states of components. It produces numeric values for current through each component, and these are quantized to values equivalent to those that would come from the qualitative component-level reasoner (values that indicate whether components are active or not). Those values can be abstracted to the functional level as states of significant components.


Lamp is ON
Function level reasoning
Active functions are deduced from component states
Ground
Component level reasoning
Resistive state is passed downwards
Component state changes are propagated upwards
Load
Ground
Load
Three-valued qualitative reasoning
Lamp is ON
Function level reasoning
Active functions are deduced from component states
Ground
Component level reasoning
Resistive state is passed downwards
Component state changes are propagated upwards
Low Load
Medium Load
Ground
Low Load
Medium Load
Multi-valued qualitative reasoning
Lamp is ON
Function level reasoning
Active functions are deduced from component states
Ground
Component level reasoning
Resistive state is passed downwards
Numerical values from simulation are interpreted and state changes propagated
0.3 Ohm
1.5 Ohm
Ground
0.2 Ohm
3.1 Ohm
SPICE numerical circuit resolution
Lamp is ON
Function level reasoning
Active functions are deduced from quantized component states
Lamp filament is ACTIVE
Component states are deduced from significant simulation results
SABER numerical circuit simulation
Ground

(a) Simulation structure for three-valued qualitative reasoning

(b) Simulation structure for multi-valued qualitative reasoning

(c) Simulation structure for numerical resistance resolution

(d) Simulation structure for SABER complex numerical simulation

Fig. 2. Use of different simulators in producing functional level reports: (a) simulation structure for three-valued qualitative reasoning; (b) simulation structure for multi-valued qualitative reasoning; (c) simulation structure for numerical resistance resolution; (d) simulation structure for SABER complex numerical simulation.

In all cases, the system provides the same functional level output from the simulation. This means that the design safety analysis results (for FMEA, sneak circuit analysis, etc.) will be directly comparable whichever simulator is used to produce the results. Comparability of results is important for a seamless design safety analysis process, as it means that changes in design results can be easily tracked as more details are added to the design. The next section gives the details of how extra results become available for each type of simulation for a typical case study. This is followed by a discussion on how designs can be tracked through the process, and the benefits of such tracking.

5. Case study—power windows

The electrically operated window system shown in Fig. 3 is a simple vehicle electrical schematic, but is representative of the type of schematics that occur in a vehicle. With the exception of ECU (electronic control unit—essentially a computer in the car), it contains the main types of components that you might see—switches, relays, wires, fuses and end effectors (motors in this case). This schematic enables the driver to operate the driver's or passenger's window in either direction (by pressing the DD_SWITCH or DP_SWITCH appropriately), and the passenger to operate their own window (by pressing the PP_SWITCH). Any operation is dependent on the ignition being powered (IGNITION_SWITCH being closed), and there are a number of fuses on the power line that should blow in different failure situations. The schematic is connected to power and to ground.

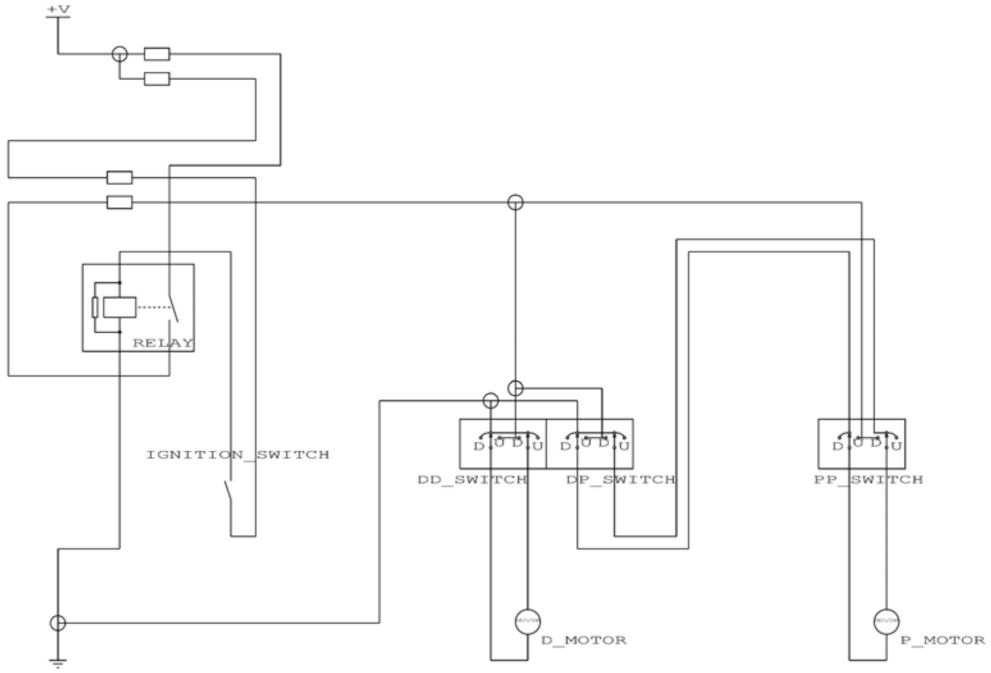


Fig. 3. Power windows circuit

This case study was one of an assortment of automotive case studies that we considered in order to assess the effectiveness and utility of performing design safety analysis repeatedly throughout the design process. The case studies were subjected to a number of tests that are typical of the analysis demanded by the automotive industry. It was determined whether the results of the simulation were able to provide the correct answers to the tests at each of the modeling stages. As well as the standard design safety analysis disciplines mentioned earlier, there are a variety of other design questions that would be of interest when examining the design of this circuit:

- *Resolving current on any bridges:* Circuits can be created with multiple paths between two nodes, and connections between those paths. A connection like this between two load paths is a bridge. To be able to analyze the behavior of the whole circuit correctly, components on bridges must be simulated correctly. A three-leveled qualitative reasoner cannot resolve bridges.

- *No power current flow through ignition switch:* Current through this part of the circuit should only be at the activation level (for powering relays), not at the power level (for powering motors, etc.). _ Voltage drop across motor under constant load: The motor requires 7 A to operate effectively. The motor has a running resistance of 1.5 V, so 10.5 V across the motor are required to achieve the 7 A.
- *Correct fuse blow under short circuit condition:* If a short circuit should occur (for example the coil in the motor shorts) the fuse should blow to stop any damage occurring.
- *Motor voltage balanced under normal operation:* When both motors are running, there should be less than a 0.5 V difference between the motors.
- *No fuse blow under motor inrush:* The resistance of the coil in a motor is very low. It is only when the motor starts rotating, and the back EMF increases, that the impedance of the motor becomes higher. The fuse should not blow while the motor accelerates and achieves its full resistance.
- *No fuse blow under stall current for 5 s:* When a motor is stalled, its resistance is decreased. The fuse protecting the system must not blow under stall conditions for at least 5 s.

The assessed capability of each type of modeling for addressing these design issues for the power windows system is shown in Table 1. This has been assessed by studying the extent to which the results from each simulator provided the desired information. In each case, a Yes means that the simulator in that column was able to produce correct results for that design issue (where correct is judged against the results produced by the engineers) for the power windows system. So, for example, neither the three-leveled qualitative reasoner nor the multi-leveled qualitative reasoner can correctly forecast which fuse will blow for a given short circuit, but both of the numerical simulations can produce this result.

It can be seen from this table that the three-level qualitative reasoner can perform the basic design safety analysis tasks for which it was originally conceived, but is incapable of any of the more detailed tasks designed to show the limitations of electrical qualitative design safety analysis. As the information available increases, more detailed simulation can answer an increasing number of the design questions. For the multi-leveled qualitative reasoning and the simple numerical reasoning based on PSPICE, this can be achieved with very little effort by replacing the three-leveled qualitative reasoner with one of the other reasoners and providing the more detailed values—the component-level modeling can be reused from the qualitative version. Where complex numerical simulation using SABER is needed, considerable effort is required in order to provide detailed and effective numerical models for components. Although, as Table 1 shows, all the design questions can be answered with this kind of modeling, it would be preferable to avoid the effort of using it except where it really is necessary.

The natural inclination of many engineers is to resort directly to SABER modeling once the qualitative reasoner fails to give detailed answers. SABER models contain complex coding, and are difficult and time-consuming to build, but where complex, transient behavior is needed, are the only models available which provide the desired results. However, in many cases, the scheme portrayed in Fig. 2(c) using knowledge of resistance values, where the engineer only needs to provide resistance values for the existing behavioral models used by the qualitative reasoner, provides equally useful answers for much less modeling effort. The simple qualitative models for component behavior used depicted in Fig. 2(a) only needs to be augmented by measuring resistance values across the specific components selected for use in order to achieve the reasoning shown in Fig. 2(c), whereas the use of SABER as portrayed in Fig. 2(d) involves detailed implementation of numerical components.

Test	3 leveled qualitative	Multi-leveled	Simple numerical	Complex numerical
Failure mode effects analysis	Yes	Yes	Yes	Yes
Sneak circuit analysis (SCA)	Yes	Yes	Yes	Yes
Design verification	Yes	Yes	Yes	Yes
Resolve current on bridges	x	Yes	Yes	Yes
No power current flow through ignition switch	x	Yes	Yes	Yes
Voltage drop across motor under constant load	x	x	Yes	Yes
Correct fuse blow under short circuit condition	x	x	Yes	Yes
Motor voltage balanced under normal operation	x	x	Yes	Yes
No fuse blow under motor inrush	x	x	x	Yes
No fuse blow under stall current for 5 seconds	x	x	x	Yes

Table 1 Capability of different simulators

6. Incremental design safety analysis

In the automotive industry, design safety analysis has typically been performed towards the end of the product design process. Where changes were made to the design after the analysis had been carried out, it was not possible to completely repeat the analysis because it would take too much time, and so engineers would estimate the effects of the change, and limit the analysis to the perceived influence of the change.

Once the design safety analysis has been automated, it is very little effort to repeat the analysis whenever a change is made to the design. However, that is not the end of the problem. The analysis is only useful if engineers look at the results, and take action on problems identified. A typical FMEA analysis might detail the effect of 500 different component failures, and so an engineer would not want to study each of those 500 results every time a small design change is made.

When the automated FMEA is initially performed, the engineer would examine all results, and take appropriate actions. When a change is made to the design (e.g. a new component added to the design), then a new FMEA report is generated. The consistency of the automated analysis results means that software can compare the results after the incremental change with the original results, and report only the differences. Results which have changed are presented to the user, along with any new results (for example, failures on components which did not previously exist). Experiments have shown that instead of being presented with 500 failure reports to consider for a single change to a circuit, the engineer might only have to study 8 or 10 failure reports [25]. This type of technology is also useful when generating diagnostics, in order to deal with the problems of variants on a design and of late design changes [14].

When extra information becomes available, such that a more detailed simulation can now be carried out, as discussed earlier in the paper, then the analysis results are presented at the same level of system functionality. The incremental FMEA facility can then be applied to detect which results have changed because of the more accurate results available from the more detailed simulation. For example, say an FMEA based on multiple-level qualitative reasoning has already been carried out and checked by an engineer. When enough detail is available to perform a SPICE-based simulation, then a SPICE-based FMEA can be carried out. The two sets of results can be automatically compared and differences between the two sets of results identified. The engineer might then be presented with the results where qualitative reasoning is unable to decide whether a short-circuit blows a fuse. These ambiguities are resolved by SPICE, and so a few numerical results will be more specific than the qualitative ones.

Use of the incremental facility has been evaluated by studying the incremental changes to the safety analysis results between the 14 different evolutions of a daylight running lights (DTRL) schematic designed by an automotive manufacturer. This was a particularly interesting design, as the manufacturer had not previously fitted daytime running lights to their cars, and so it was a design that changed more often than would be typical for an automotive schematic. The incremental results were discussed with the engineers. Some of the changes to the design were because problems had been found with the previous iteration of the design, and others were due to increasing information becoming available, such as when a supplier was established for specific components. As more detailed information became available, it became possible to use more detailed simulators to perform the design safety analysis. The software was able to identify the changed information that would be useful to the designers at each stage. The kinds of incremental changes to the FMEA report that occurred as the design changed were:

- changes to the effect of a specific failure;
- changes to the severity assessment for a specific failure;
- extra failures possible because of additional components being added to the design.

The DTRL schematic is not particularly complex, and so production of an automated FMEA report for that schematic might take a week of engineer effort without automated assistance of the type described in this paper. The automated FMEA tool could produce a similar report in seconds, but the major commitment of time with the automated tool comes during the automated examination phase—the engineers will want to examine the automatically produced FMEA report and study the significant failures in detail in order to see whether they could be avoided or mitigated. For the DTRL schematic, this process might take a few hours. Thus, it can be seen that automation of the analysis reduces a task which previously took a week to one which can be performed within half a day at the most.

Without the incremental facility described here, analysis of a repeat report on a changed version of the schematic should take as long as the first time, if the engineers study the results as seriously as they did the first time.With the incremental FMEA feature described in this paper, a repeat FMEA report for a changed version of the same circuit takes very little of the engineers' time. The engineers will usually only want to look at failure modes for which the effects have changed from the last time that FMEA was performed on this schematic. These are sorted out automatically by the system, and there are only a few of these, so the examination only takes a matter of minutes.

This incremental facility, and the fact that it works for a range of simulators,means that the implications of all design decisions can be tracked—as resistor values are decided upon or as the circuit structure changes during the design process, the effects of those decisions on the design can be seen. One could envisage a design process incorporating this facility whichwas similar to the use of regression testing in software engineering—all tests are rerun on a regular basis, and any changes to the results must be assessed and accounted for. In the same way, the technology described in this paper would allow automotive designs to be tracked on a regular basis, and could be used to prompt explanations and consideration of any changes that occur.

One possibility would be to run all design analyses each night on all systems where a change to the design has been made during the day, and provide a summary to the engineers the next day of all implications of the design decisions made during the previous day. This would minimize the detection time for any change decision that caused a new design problem, and provide efficient feedback to engineers.

7. Application to other engineering domains

This paper has shown that automotive electrical design safety analysis can been done incrementally across the design process. This section addresses the extent to which the work described can be applied outside the automotive industry, and outside the electrical domain.

The extension of this technology to electrical design safety analysis for other electrical systems containing electronic components (such as aeronautic or train electrical systems) is trivial. The technology is being used commercially by companies in each of these sectors. The representations are general enough that electrical systems in each of the markets mentioned can be reasoned about using the different levels of functionality, component and underlying electrical structure. In [4], Struss and Price survey the range of companies applying model-based reasoning for design and diagnosis. The improvements to the design safety analysis process described in this paper should work for electrical systems in all of those areas. In part, this success is due to the domain: electrical system modeling is fairly well understood, and the differences between three of the types of modeling are focused on knowledge about resistors.

Basic research on safety analysis tasks for other domains such as hydraulic, pneumatic, mechanical, or hybrid systems is also described in [4]. For domains where component-based qualitative reasoning about flow is done, separation of the qualitative reasoning from component-level reasoning can make it easier to automate reasoning, and easier to replace the qualitative reasoning with numerical reasoning. The qualitative reasoning is often fairly straightforward: the challenges come in switching between operating states of the system, and layering the information makes it easier to perform the qualitative reasoning, and easier to replace it with numerical reasoning.

Identification of the functions operating is also needed in order to focus many of the safety analysis disciplines. This means that the top-level application of functional labels is an important abstraction mechanism for safety analysis in other domains as well as electrical.

The value of the functional abstraction is well illustrated by the different reasoning scenarios shown in Fig. 2. Functional abstraction works as well with numerical reasoning as it does with qualitative reasoning. By focusing the results on what is important for design or for diagnosis, functional abstraction enables automated production of significant results as a design alters. By mapping numerical results onto qualitative ranges, this advantage is available when the design stays the same, but more information is available for the analysis. It seems likely that the same three reasoning layers could be used to apply model-based reasoning across the design process in other domains.

8. Conclusions

The work detailed in this paper has produced a model-based design safety analysis system that provides useful results throughout the design process. It has been integrated into the safety analysis software provided by one of the major electrical CAD companies. It is able to give very early feedback to engineers on the viability of their design, based on the information available before all design details have been resolved. As more detailed information becomes available, it can give increasingly accurate results. This is all achieved by separating component structural details and system function information from the information needed to perform different kinds of simulation.

At the same time, the common functional layer shared between the different simulators means that the implications of incremental changes to a design can be tracked, and the information provided to the design engineers can be minimized, only informing them of significant implications of design changes.

The work presented here has provided a framework for moving from single point design safety analysis tools to the capability to perform appropriate design analysis repeatedly and efficiently throughout the design process.

Acknowledgements

Much of this work was carried out on the UK EPSRC funded project *Whole vehicle whole lifecycle electrical design analysis*. The authors would also like to thank the anonymous reviewers of the paper for their valuable and perceptive comments.

References

[1] Potential Failure Modes and Effects Analysis in Design (Design FMEA) and Potential Failure Modes and Effects Analysis in Manufacturing and Assembly Processes (Process FMEA) Reference Manual, Society of Automobile Engineers, 1994.

[2] RODON, is owned by So¨mmer Information and Media, The RODON website is at http://www.sorman.se/products. C.J. Price et al. / Computers in Industry 57 (2006) 451–461 460 copy personal Author's

[3] SABER, is produced by Synopsis Inc., website: http://www.synopsys.com.

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

[5] N. Johnson, Driving vehicle design into a new era, Compiler (online magazine of Synopsys, available at http://www.synopsys.com/news/pubs/ compiler), February 2004.

[6] C. Price, N. Taylor, Automated multiple failure FMEA, Reliability Engineering and System Safety Journal 76 (1) (2002) 1–10.

[7] S. Savakoor, J. Bowles, D. Bonnell, Combining sneak circuit analysis and failure modes and effects analysis, in: Proceedings of the Annual Relia-bility and Maintainability Symposium, IEEE Press, 1993, pp. 199–205.

[8] C. Price, N. Hughes, Effective automated sneak circuit analysis, in: Proceedings of the Annual Reliability and Maintainability Symposium, Seattle, (2002), pp. 356–360.

[9] N. Snooke, J. Bell, Abstracting automotive system models from compo-nent-based simulation with multi-level behavior, in: Proceedings of 16th International Workshop on Qualitative Reasoning, 2002, pp. 151–160.

[10] M. Genesereth, The use of design descriptions in automated diagnosis, Artificial Intelligence 24 (1984) 411–436.

[11] J. de Kleer, B.Williams, Diagnosing multiple faults, Artificial Intelligence 32 (1987) 97–130.

[12] R. Reiter, A theory of diagnosis from first principles, Artificial Intelligence 32 (1987) 57–96.

[13] H. Milde, T. Guckenbiehl, A. Malik, B. Neumann, P. Struss, Integrating model-based diagnosis techniques into current work processes: three case studies from the INDIA project, AI Communications 13 (2000) 99–123.

[14] C. Price, Incremental automated diagnostics, in: Proceedings of the AAAI Spring Symposium on Information Refinement and Revision for Decision Making: Modeling for Diagnostics, Prognostics, and Prediction, Palo Alto, March 2002.

[15] F. Cascio, L. Console, M. Guagliumi, M. Osella, A. Panati, S. Sottano, D. Theseider-Dupre´, Strategies for on-board diagnostics of dynamic auto-motive systems using qualitative models, AI Communications 12 (1/2) (1999) 33–43.

[16] C. Price, N. Taylor, Multiple fault diagnosis using FMEA, in: Proceed-ings of the AAAI97/IAAI97 Conference, Providence, RI, (1997), pp. 1052–1057.

[17] D. Ward, C. Price, System functional safety through automated electrical design analysis, in: Proceedings of the SAE 2001 Transactions, Section 7, vol. 110: Electronic and Electrical Systems, Journal of Passenger Cars (2001) 341–347.

[18] L. Chittaro, G. Guida, C. Tasso, E. Toppano, Functional teleological knowledge in the multi-modeling approach for reasoning about physical systems: a case study in diagnosis, IEEE Transactions on Systems, Man and Cybernetics 23 (6) (1993) 1718–1751.

[19] M. Lee, Qualitative circuit models in failure analysis reasoning, Artificial Intelligence 111 (1999) 239–276.

[20] N. Snooke, Simulating electrical devices with complex behavior, AI Communications 12 (1/2) (1999) 45–59.

[21] C. Price, Function directed electrical design analysis, Artificial Intelli-gence in Engineering 12 (4) (1998) 445–456.

[22] J. Keown, OrCAD PSpice and Circuit Analysis, 4th ed., Prentice-Hall, 2000.

[23] M. Lee, J. Bell, G. Coghill, Ambiguities and deviations in qualitative circuit analysis, in: Proceedings of the 15th International Workshop on Qualitative Reasoning, 2001, pp. 51–58.

[24] P. Struss, Automated abstraction of numerical simulation models: theory and practical experience, in: Proceedings of the 16th International Work-shop on Qualitative Reasoning, 2002.

[25] C. Price, Effortless incremental FMEA, in: Proceedings of the Annual Reliability and Maintainability Symposium, 1996, pp. 43–47.

Chris Price is currently the head of the Department of Computer Science at the University of Wales Aberystwyth. He is a Fellow of the British Computer Society. He has worked in model-based and qualitative reasoning for the last 20 years, publishing regularly in this field, and is a director of the EC-funded European Network of Excellence in model-based and qualitative reasoning. His current research interests include model-based diagnosis of unmanned aerial vehicles, using qualitative reasoning in education, and learning qualitative models.

Neal Snooke received an Honours degree in micro-electronics and computing in 1990 and a PhD in wavelet-based image compression in 1994. While working as a research associate in the model-based systems group he was involved in the development of the AutoSteve automotive electrical FMEA tool and acted as research director of a successful spin-off company. He is currently a lecturer at the Department of Computer Science at Aberystwyth, specillizing in network technologies and telematics. His research interests include model-based and qualitative reasoning with application to automated design analysis tools for electrical, electronic, network-based, and embedded systems and software.

Stuart Lewis graduated in software engineering, and worked in the area of qualitative reasoning research under the guidance of Professor Price and Dr. Snooke. Much of his work centered around the design lifecycle, and tools that support analysis of electrical circuits. More recently he has been working in the area of open-access repositories for research output and is currently leading a research project investigating aspects of such systems.