AutoSteve: Automated Electrical Design Analysis

Chris Price[1]

Abstract. AutoSteve performs automated electrical design analysis based on qualitative simulation and functional abstraction. It is the first commercial product capable of performing these tasks for complex automotive systems. It has been deployed at automotive manufacturers for several years, and produces FMEA and sneak circuit analysis reports much more quickly and consistently than they could be produced without its assistance.

1 PROBLEM DESCRIPTION

There is a trend towards increasing complexity of electrical/electronic systems in modern vehicles, caused by pressures on automotive manufacturers to improve efficiency, safety and vehicle features. At present, electrical systems account for approximately 10% of total vehicle cost, and include some of the most challenging aspects of the vehicle design. Further complications are added by variants in vehicle configuration, where installed options can make the possible interactions between vehicle systems dependent on the configuration chosen by the customer. It is a major challenge to assess the safety and reliability of such systems as early as possible in the design process.

In order to make sure that possible shortcomings of a design will be detected, a number of design analysis techniques have been developed.

- FMEA. Failure mode and effects analysis [1] considers the effect on an overall product of any (usually single) failure of part of the product.
- FTA. Fault tree analysis [2] highlights the combinations of failures that can affect the safety of a design.
- Design verification. Given a formal description of the legal states in which a system can be, it is possible to analyse the operation of the design to ensure that the device cannot enter any illegal states.
- Sneak circuit analysis [3]. This identifies any unexpected interactions between systems within a product.
- Yellowboarding. Unlike most of the design analysis techniques mentioned here, this usually involves construction of a physical prototype. To ensure that the electrical systems of a vehicle are designed correctly before constructing a complete prototype of the vehicle, the electrical systems are pegged out on a large board and tested against expected behaviour.

Because designs change during development, and the analysis takes a lot of engineer effort, it is often performed late in the design process once the engineers are relatively confident that the design is frozen. By this time, any changes needed will be expensive to perform and will slow down the release of the vehicle.

[1]Department of Computer Science, University of Wales, Aberystwyth, SY23 3AZ, UK and FirstEarth Limited, The Mill, Mill Street, SY23 1JB, UK.

With the exception of yellowboarding, each of these analyses is based on engineers calculating the behaviour of the overall system in different states (under changing inputs and failure states). The necessary work is repetitive, error-prone and takes a long time (a large FMEA analysis can take several months), but has resisted automation. Numerical modelling tools such as Saber provide some help with single simulations, but need a level of detail which means that they can only be performed late in the design lifecycle, and provide results which are difficult for the user to interpret.

This paper describes AutoSteve, a system which automates many of these design analyses, and which provides a foundation for automating them all. AutoSteve uses qualitative reasoning to provide quick, early and accurate analysis reports.

2 APPLICATION DESCRIPTION

The basic capabilities needed in order to automate these design analysis tasks are faithful simulation of electrical system operation, and interpretation of the results of simulation. Numerical simulation has failed to provide these, because the simulations have been too onerous to construct, and too difficult to interpret.

AutoSteve uses two AI technologies to achieve automated design analysis:

- qualitative simulation
- functional abstraction

Each of these technologies is described, followed by the detail of how they are used in AutoSteve.

2.1 Electrical qualitative simulation

The main intuition behind qualitative simulation is that much of the reasoning done by engineers is done at a qualitative level. Tracing the behaviour of vehicle schematics is mostly done at the level of current flow, rather than needing to calculate the exact current to two decimal places. This is all that is possible early in the design life-cycle, when exact values for resistors are not known. When exact values are needed in order to deduce correct results, that fact can be highlighted by a qualitative simulation, and examined in more detail later in the design life-cycle.

Qualitative simulation can be carried out for circuits where only a qualitative description of component behaviour is known. Such qualitative descriptions can cover many real components (for example, only one switch description might cover many similar types of switch), and so are highly reusable.

The description of component behaviour that is needed for each type of component has three separate aspects:

Terminals: Terminals are the inputs and outputs for the component. They are the ports where other components can be connected to this component.

Internal topology of component: The functionality of the component is determined in terms of links between terminals.

These links can include logical resistors, where the resistance value can change depending on the state of other parts of the component.

Dependencies: Dependencies define how the values of the internal resistors of a component change as the state of the other parts of the component change.

Example of behaviour for a switch: A switch has two terminals. The terminals can be regarded as joined by a variable resistor whose value depends on the state of the switch. When the switch is open, then the resistor has infinite resistance. When it is closed, the resistor has zero resistance.

Example of behaviour for an open relay: An open relay is composed of a coil, and a switch whose state depends on the state of the coil. When current flows through the coil, the switch is closed, otherwise it is open. Such a relay has four terminals, two to the coil, and two to the relay switch. The coil will be a fixed resistor, and the switch resistor will be variable and depend on the state of the coil. When the state of the coil is Active, i.e. current is flowing through it, then the value of the switch resistor is zero because the switch is closed. When the state of the coil is Inactive, i.e. no current is flowing through it, then the value of the switch resistor is infinite as the switch is open.

When the structure of a circuit is drawn within an electrical CAD tool, a netlist can be extracted and used with the component descriptions to simulate the circuit. AutoSteve uses CIRQ [4] to analyse where current is flowing through a network of resistors. Given a circuit to simulate, and the initial state of each component in the circuit, the simulation controller will perform the following steps:

- Build a resistive network from knowledge of the component states and the connections between components.
- Pass the resistive network to CIRQ, and get back details of where current is flowing in the network.
- Use the details of the current flow to identify components whose internal state has changed.
- If any components have changed state, repeat from step 1, else terminate.

For several of the types of analysis mentioned earlier, the simulation must also be able to deal with the behaviour of a failed component. This is achieved by substituting the description of the correct behaviour of a component with a description of its behaviour when it has a specific failure, for example, a relay might have failed because its coil was burned out. The failed behaviour in that case would be that the switch in the relay never closed, so the value of the switch resistor was always infinite.

The result of qualitatively simulating a circuit is a changing set of values for each component in the circuit as the inputs to the circuit (switches, sensors, ECU states) are changed. If this result is presented as a list of components and their states, it can be very difficult to comprehend. This is especially true as circuit complexity increases. Functional abstraction is used to interpret the results of simulation.

2.2 Functional abstraction

Electrical systems analysed in AutoSteve can have hundreds or even thousands of components. This would give many thousands of values for component states during a simulation. This amount of information is far too detailed to expect an engineer to look at all result values. The qualitative simulation, like numerical simulation, provides no way of abstracting the important information from the morass of detail.

There is a strand of AI known as functional reasoning [5, 6] which characterises the significant overall behaviour of a system in terms of the functions that the system performs. AutoSteve uses functional labels [7] to identify the important attributes of a system or device. Typically, the significant overall behaviour of a system can be characterised by a few such labels. Examples of functional labels for specific car subsystems might be:

External lighting system:

High beam
Low beam
Sidelights
Stop lights
Right indicators
Left indicators
Fog lights
Reversing lights

Central locking system:

Doors locked
Doors open
Doors locking
Doors opening
Doors deadlocked

In order to use functional labels to simplify and interpret the behaviour of the qualitative simulation, it is necessary to be able to identify when the functions are being carried out in the simulation. The presence of each function can be identified from the states of key components.

Functional labels have a high level of reusability between different implementations of a subsystem, provide an appropriate level of abstraction for interpreting circuit behaviour, and are a mechanism for producing analysis results that are at the correct level for presenting to engineers.

Within AutoSteve, functional labels have a range of different uses:

- as links to recognise and interpret circuit activity,
- as a basis for assigning severity and detection values to each possible failure,
- as a way of selecting English language failure effect reports,
- as a basis for deciding when sneak effects are occurring
- as labelling for state charts during design verification

Functional labels can also be used to focus numerical simulation, as will be discussed in the further work section.

2.3 Generating Design Analysis Results

Failure mode and effects analysis

Figure 1 shows an example of failure mode and effects analysis produced by AutoSteve in the following way:

- Simulate the correct behaviour of the circuit through a set of input changes (trying out all the possible operations of the circuit).
- Abstract the results of the simulation using functional labels to obtain a set of input/function mappings.
- Repeat the simulation for each possible failure on each component of the circuit and abstract the results.
- Compare the abstracted results and report any differences.

Each possible failure generates one row of the FMEA report, and the results can be reordered by failure mode if preferred.

Failure	*Potential Failure Mode*	*Potential Failure Effect*
Horn relay J4 has failure switch stuck open.	When Main_Crash_Sensor was set to detected, the "horn sounds" function was not achieved. Finally, regardless of any event change, the "Frontal bag and belts" function and the Warning Lamp illuminated" functions were never achieved.	Possible death of occupants because of airbag failure. Warning lamp fails to illuminate.

Figure 1: Extract from one row an of airbag FMEA report

Sneak circuit analysis

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

Sneak circuit analysis is the process of identifying and eliminating such sneak paths where they might occur. Where a wire is allowing current to flow in an unexpected direction, this can often be prevented by the addition of a diode to the design, but cost, weight, and reliability considerations mean that extra diodes should not be added to the design unless they are really needed.

Further information is required in order to perform sneak circuit analysis in AutoSteve. For each of the functions of the subsystem, it is necessary to declare the legal combinations of inputs under which that function will be active. This is achieved with a simple interface window where the engineer enters the information.

All possible switch combinations are simulated by AutoSteve. Sneak circuits are detected as a function operating under an illegal set of inputs, or not operating under a legal set of inputs. This is more efficient and more accurate than other attempts at automated sneak circuit analysis, which detect current flowing "the wrong way" in components. For classic documented sneaks, it detects all possible sneak combinations, and does not generate any spurious problem reports.

Other types of design analysis

A research prototype for design verification [8] has been achieved using the mechanisms described in this paper, but has not yet been integrated into the AutoSteve system. Essentially, it generates a state chart of all possible states of the subsystem being simulated, and compares it with a state chart containing the original specification for the subsystem. There are several reasons why this work has not moved beyond research as yet. The most practical one is that automotive engineers are not producing state chart specifications for the overall required behaviour of systems as part of the design process at present. One might imagine it becoming part of standard practice, at least for safety critical systems, in the future. At that point, a design verification tool would become commercially viable.

One of the uses of fault tree analysis is to compensate for the shortcomings of manual FMEA. It is used to highlight all of the combinations of failures that will make a particular unwanted event occur. For example, such an event might be a vehicle's airbag firing when it should not. Alternatively, it might be to identify when the airbag will fail to fire. It is then possible to calculate an overall figure for how likely it is that unwanted event will occur. Engineers calculate the dependencies in the fault tree by hand. The multiple failure FMEA work described in [9] and included in AutoSteve provides all of the information that is needed to decide what combinations of failures can cause the unwanted event to occur. In addition, as vehicles become more complex, with ECUs programmed to mitigate the effects of known failures, it is likely to calculate the effects of a combination of failures more accurately than an engineer simulating circuit operation in their head.

Late in the design process, before fitting a new design for a car's electrical systems into a prototype vehicle, engineers will peg out the wiring harness and associated electrical devices onto a board. The yellowboard version of the electrical systems can then be used to test that all devices work as expected. This is typically done using a script for the changes to be applied to each system, and the expected (functional) results. Virtual yellowboarding can be achieved with AutoSteve as soon as the electrical circuit has been drawn in a CAD tool. A scenario for each system is set up (as for FMEA), and a correctly working version of the circuit is simulated through each step of the scenario and the results are reported to the user. This is not a replacement for physical yellowboarding, but can be done cheaply much earlier in the design, and any problems found in the design can be eradicated before the physical prototype is even built. An additional benefit is that running of the scenario on a virtual prototype reveals any errors in the yellowboarding script, and so it saves time later during physical yellowboarding.

2.4 Integration with conventional systems

In order to make it as easy as possible for the engineers to use AutoSteve, it needed to be linked to the tools which they already used, and share their look and feel. For electrical design analysis, that means that AutoSteve needs to be integrated with the electrical CAD tools that engineers use to draw schematics. AutoSteve is implemented as an extension to the CAD tool, having its own drop-down menu within the CAD tool's menu structure.

It takes lists of components and their connectivity directly from the CAD tool, and more importantly, it can colour the schematic within the CAD diagram to show circuit activity using the results of simulation. This is important, because the main reason for performing design analysis is for the engineers to understand better the circuits and the possible implications of problems with the circuit design.

In addition to being able to observe which parts of the circuit are active by colouring wires, direction of current flow is indicated by arrows. This can be important. In a headlamp circuit, a particularly nasty set of results were achieved when a local ground to the left headlamp cluster was lost. Instead of the expected two lamps being illuminated, a total of 8 lamps were lit. The engineer's initial impression was that the modelling was in error. Eventually, close examination of the direction of current flows helped the engineer understand that it was a rather nasty sneak effect involving currents running back to common fuses and switches that were not powered - all because of the lost ground. These visualisation features are integrated with the other design analysis techniques, so that FMEA,

for instance, can set up the circuit for simulation with specific faults induced on components, and visually demonstrate the effect of that failure on the circuit. In the same way, sneak circuit analysis can illustrate the sneak conditions by setting up the visualisation so that the sneak path is clearly coloured.

AutoSteve was originally implemented with all information stored in a directory file structure. As the amount of information increased, this became less practical as a solution, and so the file structure was transformed into an SQL database capable of holding the large amounts of information generated by design analysis, as well as the library of models for components, and other associated information.

3 APPLICATION BUILDING

A first observation is that AutoSteve is built on top of ten years of research into performing automated design analysis at the Department of Computer Science, University of Wales Aberystwyth. The first "Flame" system prototype was achieved in 1991, following a study of engineers carrying out FMEA, and experimentation in how their reasoning might be reproduced [10]. That research in its turn built on the experience of previous qualitative electrical systems experimentation over the previous decade [11,12].

Development of a version of the FMEA system linked to a commercial CAD tool (TransCable) was funded by Ford Motor Company and built with four man years of effort during 1996/7. The difficult work had all been completed during research projects, and so the development was done using the waterfall model, with a clear requirements specification of the expected end product written at the start. AutoSteve 1 was applied to all the electrical systems of a new car design, and the lessons learned were documented [13]. Extensive application of AutoSteve 1 to car systems showed that the representation of complex electronic components using the kind of dependency description shown earlier took more effort than was necessary, and was incapable of reproducing the behaviour of components with time-dependent behaviour.

A further research project at the University experimented with different kinds of component representations, and produced research prototypes of sneak circuit and design verification tools. State chart based component descriptions [14] make it much easier to describe the behaviour of complex components. A good example of such a component is an ECU within a central doorlocking circuit, where the ECU might have to detect that the circuit was locking the doors, and reset all the doors as unlocked if the locking process was not completed within a few seconds. To describe the behaviour of this component as a set of dependencies between resistors takes several hundred lines of dependency expressions, whereas it can be described as a state-chart containing half a dozen linked boxes.

FirstEarth Limited was formed in 1997 to support and sell the design analysis tools produced by research in the Department. In 1999, it chose to add state chart based components to AutoSteve, to implement the sneak circuit work, and to begin linking AutoSteve to different CAD tools. After another five man years effort, it produced AutoSteve 2 at the end of 1999.

As a commercial product supported on several different platforms, it has become difficult to separate adaptive maintenance from further development work. However, the development figures for AutoSteve total around 9 man years, on top of 12 man years of research effort.

These development figures contain a useful lesson for university researchers. We thought we had a tool that industry could use after about 6 man years of the research effort – there is a large gap between a useful research prototype and a commercial product. On the positive side, interaction with industry has driven the research to look at more challenging problems.

4 APPLICATION BENEFITS

AutoSteve 2 has been adopted by Ford Motor Company internationally as part of its process for developing electrical systems, and is being adopted by other automotive companies. This has been a long adoption process, trying it out on specific car model developments, and assessing the usefulness of the results before coming to a decision. The benefits of AutoSteve which have led to this decision are:

It performs design analysis at the right point in the design process. Numerical simulation based analysis needs more information and more engineer effort than is available early in the design process. AutoSteve asks for appropriate amounts of information, and provides usable results early enough in the process that it is still relatively inexpensive to fix any problems.

It generates an FMEA or sneak circuit report several orders of magnitude faster than can be achieved without it. Complex

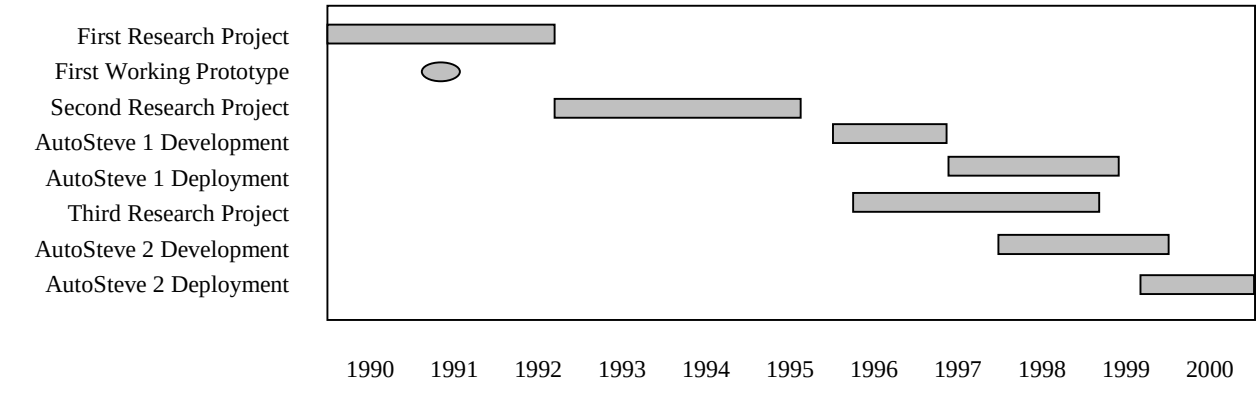

Figure 2: TimeLine for AutoSteve development

systems, where it might take several months to produce an FMEA report manually, can be analysed in less than a day. Car companies are trying to reduce the time needed to produce a new car model, and AutoSteve helps with that.

It reduces the need for physical prototyping. Building physical prototypes is a very expensive business, and automotive companies are trying to reduce the number of prototypes which are built. AutoSteve does not take away the need to build physical prototypes, but it helps to reduce the number of prototypes which are needed.

It enables the engineer to perform design analysis with as little effort as possible, while still providing them with an understanding of the systems they are designing. Conservative estimates for the introduction of new technology indicate that the benefit/effort ratio needed for engineers to adopt new tools has to be around 10:1. Industrial experience with AutoSteve has shown that it has the balance correct between the benefits of automating much of the generation of analysis reports and the effort needed to build qualitative models and functional descriptions of systems. This has been achieved by making the models and descriptions highly reusable, and by linking all of the tools closely to the CAD tools used by the engineers.

It produces consistent FMEA reports. This is an advantage in several ways. It allows manufacturers to compare different FMEA reports, knowing that they have all been reported on the same scale. In addition, the same problems will always be reported in the same way. This means that the FMEA results can be used to generate all the component failures which could cause a specific failure mode. This is useful both for building diagnostic systems, and also for performing criticality analysis.

Traditionally, engineers have performed design analysis by hand – performing mental simulation and calculating the effects of failures on a design, or trying to detect unforeseen interactions within a design. This is a long, time-consuming, process and prone to inconsistencies. The automated design analysis performed by AutoSteve enables the analysis to be performed early in the lifecycle and in a significantly shorter time. Instead of replacing staff, AutoSteve supports their effort and enables more detailed and consistent analysis. Savings are made in improved detection of potential problems, leading to less design rework and less recalls of vehicles due to electrical problems.

5 FUTURE PLANS

Qualitative design analysis is efficient in engineer effort and provides excellent feedback to engineers very early in the design process. It can be complemented by a numerical design analysis much later in the design process, and much of the information that was set up for the qualitative design analysis can be reused.

Over the past year, FirstEarth have been developing a second set of design analysis tools using a numerical simulator instead of a qualitative simulator. These tools are presently being field tested. They take the functional labels, sneak circuit descriptions and other information set up for AutoSteve, and reuse it later on in the design process, once exact values for components and voltage drop values are known. Results from the numerical simulator are mapped on to qualitative bands, and can then be abstracted to function operation in the same way as happens for the qualitative simulator.

Thus, AutoSteve and the new numerical design analysis tools straddle the design process, with AutoSteve providing early feedback, and the new tools verifying the results from AutoSteve are still correct once more detailed information is available.

Another challenge being addressed by the University is concerned with the size of system which can be analysed by these methods. Both qualitative and numerical simulation tends to be done at the subsystem level. However, qualitative simulation can operate on complete car electrical systems, and we are investigating how whole car simulation might affect the way in which design analysis is carried out in the automotive industry.

ACKNOWLEDGEMENTS

This work has been carried out by the University of Wales Aberystwyth and by FirstEarth Limited. The University have been supported in this work by Ford Motor Company, and by the UK Engineering and Physical Sciences Research Council (grants numbered GR/L20542, GR/H96973, and GR/N06052).

REFERENCES

1. W. Jordan, Failure Modes, Effects and Criticality Analyses. in Proceedings of Annual Reliability and Maintainability Symposium, 30-37, IEEE Press, 1972.

2. W. Lee, D. Grosh, F. Tolman, C. Lie, Fault tree analysis, methods and applications: a review, IEEE Transactions on Reliability, vol. R34, 194-302, 1985.

3. D. S. Savakoor, J. B. Bowles, R. D. Bonnell, Combining sneak circuit analysis and failure modes and effects analysis, in Proceedings of Annual Reliability and Maintainability Symposium, 199-205, IEEE Press, 1993.

4. M. H. Lee, Qualitative Circuit Models in Failure Analysis Reasoning, Artificial Intelligence, 1999.

5. Sembugamoorthy, V. & Chandrasekaran, B. Functional Representation of Devices and Compilation of Diagnostic Problem-solving Systems. in Experience, Memory and Reasoning. eds. Kolodner and Riesbeck, Lawrence Erlbaum, 1986, pp. 47-73.

6. Sasajima, M.; Kitamura, Y.; Ikeda, M.; Mizoguchi, R. FBRL: A Function and Behavior Representation Language. in Proceedings IJCAI-95, 1995, pp. 1830-1836.

7. C. J. Price, Function Directed Electrical Design Analysis, Artificial Intelligence in Engineering 12(4), pp 445-456, 1998.

8. C. J. Price, A. G. McManus, N. Snooke, Automated Design Verification Of Automotive Electrical Circuits, in Procs 1999 Vehicle Electronics Systems Conference, Coventry, 1999.

9. C. J. Price and N. S. Taylor, "FMEA for Multiple Failures", Proc. Ann. Reliability and Maintainability Symp., 1998, pp 43-47.

10. C. J. Price, J. E. Hunt, M. H. Lee and A. R. T. Ormsby, A model-based approach to the automation of failure mode effects analysis, Proc Instn Mech Engrs, Part D: The Journal of Automobile Engineering, vol. 206, pp285-291, 1992.

11. J. S. Brown, R. Burton, J. de Kleer. Pedagogical and Knowledge Engineering Techniques in SOPHIE I, II and III, in "Intelligent Tutoring Systems", D. Sleeman and J. S. Brown (eds.), pp227-282, Academic Press, New York, 1982.

12. D. Bobrow (ed.), Qualitative Reasoning About Physical Systems, North-Holland, 1985.

13. N. Snooke and C. J. Price, Challenges for Qualitative Electrical Reasoning in Automotive Circuit Simulation, in Proceedings of 11th International Workshop on Qualitative Reasoning QR-97, pp175-180, Cortona, Italy, June 1997.

14. N. Snooke, Simulating Electrical Devices with Complex Behaviour. *AI Communications* 12 (1,2), 45-59 1999.