Formal Verification of Safety Automation Logic Designs

Janne Valkonen[1], Matti Koskimies[2], Kim Björkman[1], Keijo Heljanko[2], Ilkka Niemelä[2], Jari J. Hämäläinen[1]

[1]Technical Research Centre of Finland (VTT), P.O. Box 1000, FI-02044 VTT,
Phone +358 20 722 111, Fax +358 20 722 6027, janne.valkonen@vtt.fi, kim.bjorkman@vtt.fi,
jari.hamalainen@vtt.fi

[2]Helsinki University of Technology (TKK), Department of Information and Computer Science, P.O. Box 5400,
FI-02015 TKK,
Phone +358 9 4511, Fax +358 9 451 3369, keijo.heljanko@tkk.fi, ilkka.niemela@tkk.fi

KEY WORDS safety automation, I&C, model checking, formal verification, NuSMV

ABSTRACT

In safety critical processes, especially in nuclear power plants, the new digitalized automation (I&C) systems have brought out new needs for safety evaluation. The programmable digital logic controllers can perform complicated control tasks and, thus, their comprehensive verification against safety requirements is a difficult task. Model checking is a promising approach that enables complete verification of a logic design when a finite state machine model of the control logic is available. The paper describes the use of model checking for the verification of an arc protection system and summarizes experiences of utilizing model checking in automation design and verification. For the verification of the arc protection system, it was necessary to model the overall design of the system and its operation environment. The environment model could be kept relatively simple while covering the essential behaviour of the environment. The results show that it is possible to reliably verify the presence of a desired or the absence of an undesired behaviour of the system. The possibility of complete verification makes model checking different from simulation based testing where only selected schemes can be simulated and one can never be sure that all the possible sequences are examined.

1 INTRODUCTION

The verification of safety I&C designs still relies heavily on subjective evaluation. Formal methods have been studied, but often they are only used for certain tasks as indicators of possible problems. Model checking is a promising approach that at least theoretically enables complete verification of the system safety requirements, which is not possible with traditional simulation methods. A detailed dynamic model of the process and automation can be utilised in simulation-based analysis. However, model checking is based on a so-called state machine model of the control logic and the essential surrounding systems. Model checking algorithms that analyse the state machine model are applied for verifying the safety requirements one by one. Model checking can also handle delays and other time-related operations, which are crucial in many automation system logic designs and challenging to design and verify.

This paper describes some of the results of the research project MODSAFE (Model-based safety evaluation of automation systems) in the Finnish Research Programme on Nuclear Power Plant Safety 2007–2010 (SAFIR2010) /10/. As the main case, this paper discusses the model checking based verification of an electric arc protection system with selective multi-zone protection.

2 MODEL CHECKING METHODOLOGY

The traditional way of validating safety critical automation systems has relied heavily on two standard techniques of testing and simulation. However, these techniques frequently do not scale at the rate of the system's size growth. Model checking /9,7/ is a set of methods for analysing whether a model of a system fulfils its specification by examining all of its possible behaviours. Good introductory books on model checking are /1,6/. The employed models are usually finite state models such as finite state machines, but can contain very large state spaces that the model checking tools are tailored to efficiently explore with efficient algorithms. In this paper, state machine models contain state variables holding the current state of the modelled system, as well as holding any state needed for modelling the environment the analysed system is interacting with. In verifying such models, we employ much of the technology currently being applied for circuit validation applications, such as microprocessor validation tools. For more background on model checking in the context of NPP I&C, see /13/.

In model checking, the model analysis can be made fully automatic with computer-aided tools. In our case, the models were written in the input language of the NuSMV model checker /5/. The specifications the models are required to fulfil were expressed in linear temporal logic /6/. Now, given a model and its specification as input, a model checker decides whether the system violates its specification or not. If none of the behaviours of the system violate the given specification, the (model of the) system is correct. Otherwise, the model checker will automatically give a counterexample execution of the system model demonstrating why the property has been violated.

In symbolic model checking employed in this paper, the main idea is to represent the behaviour of the system very compactly in a symbolic form. There are several variations of symbolic methods. The most well-known is the use of a data structure called ordered binary decision diagrams (OBDDs), which are a canonical representation of Boolean functions /3,4/. This allows a much more efficient memory usage and often also faster examination of the reachable states of the system than with methods representing each one of the reachable states of the system explicitly.

3 ARC PROTECTION SYSTEM

We have studied how model checking can be used to verify the design of an electric arc protection system. For the case study, we considered a rather involved setting where selective multi-zone protection is designed for a typical power distribution set-up. The system consists of a master unit, overcurrent sensor units, and light sensor units. Sensors are installed into the protected system and connected to the master unit via optical cables. The master unit collects the alarm signals from sensors, and when necessary, launches circuit breakers which close the power feed from the protected device leading to termination of the electric arc. The master unit is based on a Programmable Logic Controller (PLC) so that one can freely design and program the tripping logic according to the protected system and the protection required for it. This provides the possibility for selective tripping: the protected system can be divided into several protection zones with different tripping conditions. Figure 1 shows the switching diagram of the system design that we are considering, for more details, see /12/.

The protected system is divided into three distinct protection zones. For all of these there is a zone-specific tripping condition which causes tripping of the circuit breakers. The protection system is designed to operate with each protection zone so that there are two levels of backup breakers. That is, if the primary breakers are broken, the protection system tries first to cut down the power feed only from the main power feed that is closest to the alarming zone. If the alarm is still on (which might result, for example, if the connecting breaker C was broken), then the power feed will also be cut in the other main power feed, which will lead to cutting down the power feed in the whole system. Figure 2 shows a tripping logic of the design under verification with four configurable delay parameters D1-D4. The goal of the design is to guarantee that:

- The installation of the sensors and the tripping logic should conform to the specified tripping conditions (see /12/).
- The backup breakers should not be tripped unless necessary.
- Existence of an electric arc in the protected system leads eventually to shutting down the power feed to the protected system.

In order to verify these properties, we need to model the essential features of the arc protection system and the protected system and then formalize unambiguously the requirements.

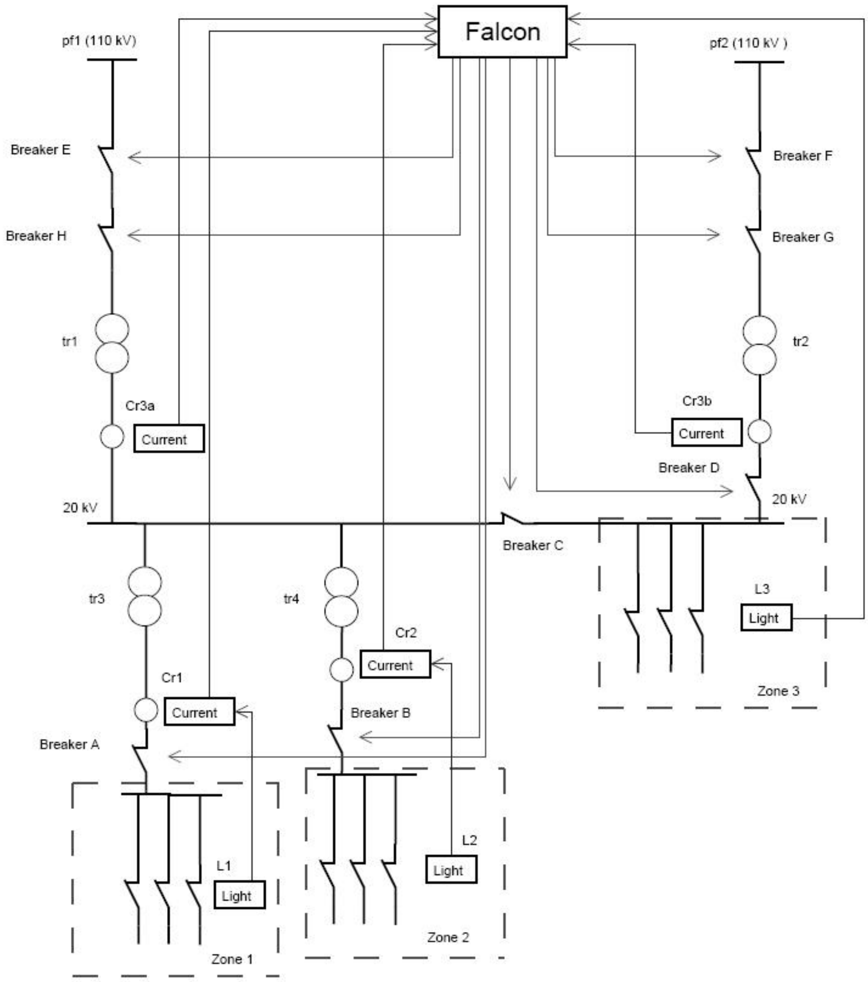

Figure 1 The switching diagram of the system design in the arc protection case.

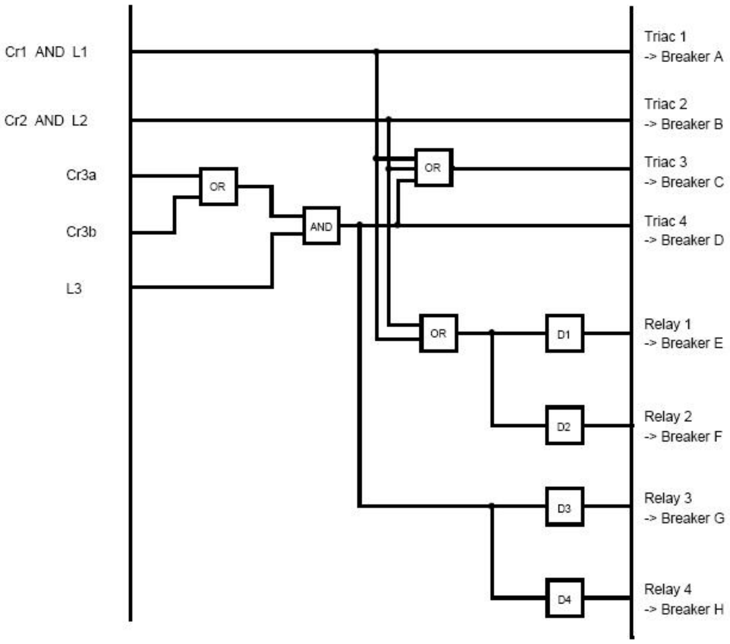

Figure 2 The tripping logic in the arc protection case.

The assumptions of the functional and behavioural properties of the system made in the modelling are described in detail in /12/. Basically the system is assumed to operate in discrete time steps. An environment model that includes all the potential sensor alarms under each configuration of the circuit breakers is also needed. Thus, it is possible to systematically build a NuSMV model that captures the essential behaviour of the protected system and the arc protection system. The delays are modelled with modules acting as counters that work as building blocks for the physical parts where needed. For a detailed description of the NuSMV model, see /12/. The model contains five adjustable parameters, which are the physical activation delay A of the circuit breakers, and the delay parameters D1, D2, D3, and D4 of the four different delay gates of the tripping logic (see Figure 2). The parameters are defined as positive integer values that correspond to milliseconds in real life. For the experiments, a standard PC and NuSMV ver. 2.4.2 with BDD-based LTL model checking were used. Depending on the values of the adjustable parameters the state space of the system model varies between $3.0 \cdot 10^{21}$ and $2.4 \cdot 10^{29}$ states (size of the state space can be calculated as the product of the value ranges of each state variable of the model). The verification times ranged from 1 min to 3.5h with different values of parameters A and D1-D4, see /12/. If the delay parameters for a certain activation delay A are chosen to be too small, all the properties are not valid anymore. In this case, the NuSMV model checker returns a counterexample for each property that is violated.

4 EXPERIENCES OF MODEL CHECKING IN AUTOMATION DESIGN

The suitability of model checking for various kinds of verification problems has been evaluated by studying several other example systems (case studies). One of the cases was the emergency cooling system of a nuclear reactor. The purpose of the system is to cool the reactor core when the normal cooling systems are unavailable. When the water level in the reactor container gets too low, water is pumped in as long as the water level gets back to a safe level. The system’s control logic and the most relevant physical parts were modelled as a finite state machine to test the suitability of model checking and to verify the safety properties of the system. The physical parts included in the model were several pumps, valves, and the water level of the reactor container. The input and output signals were connected to each other through the modelled physical parts to get the feedback loop to the system. The objective was to verify the correctness of the system’s logical functions and test different approaches to modelling. Several properties of the system were verified with model checking and no erroneous behaviour between the system model and its specification was found. However, the potential and power of the model checking method were clearly demonstrated. For further information of this case study, see /12/.

Another analysed system was a stepwise shutdown logic /2/. It is used for the stepwise control of an industrial process towards the normal operating state in case of disturbances. The purpose of the system is to reduce the possibility that the process enters a state where the more complicated actual shutdown function is launched. The system design consists of logic gates and a timed loop to make the control stepwise, i.e. the system is driven towards a safer state for a certain period after which it waits another period and continues this cycle as long as necessary. The safety logic of the system had two optional designs that were modelled and verified with two model checking tools, NuSMV /8/ and UPPAAL /11/. The performance and the applicability of the tools were analysed and compared. Both tools were found useful in verifying the system’s basic safety properties. In addition to verifying the correct behaviour of the design, NuSMV was successfully used to analyse whether the single failure criteria based on several different failure models were satisfied /2/. The failure scenarios were created by combining the following properties: the failures were detected or they remained undetected, failed input signals were given non-deterministic values or they kept their previous values, and input signals might fail or recover at any time step.

The experiences of model checking have been very encouraging. The analysed case studies demonstrate how small subtle changes in the design may lead to unexpected errors that are hard to detect without exhaustive model checking techniques. For example, a system having 10 inputs, few delay and time pulse components and a feedback loop may easily have a state space greater than 10^{10}. Manual inspection or exhaustive testing of that kind of design is nearly impossible but making a formal model is rather straightforward. Computation times of model checking such a system with a state space of 10^{10} are typically less than a few minutes. If the design is modified, the formal model of the system can easily be updated and model checked again with reasonable effort.

Both of the employed model checking approaches (NuSMV and UPPAAL) are able to verify moderate size safety logics. However, the challenge in validating digitalized safety I&C systems is the combination of timing aspects with control logic in a setting where requirements need to be verified in all possible combinations of a large number of input variables.

5 CONCLUSIONS

Model checking tools typically offer a finite state machine based modelling language for modelling the system to be verified, a specification language based on temporal logic for expressing the properties to be verified, and a set of analysis tools to check that the system satisfies the given properties. We employed a state-of-the-art open-source model checking system, NuSMV, and with reasonable effort we were able to (i) model a realistic system on an adequate level, (ii) formulate required safety properties in the specification language, and (iii) perform a full verification of the properties using the NuSMV system. This indicates that the current model checking techniques are applicable in the analysis of safety I&C systems. We have also conducted other similar case studies /12,2/ where designs of interesting safety I&C systems have been verified using model checking techniques.

These results show the potential and power of model checking and give a good basis for future research. After making the model, it is rather easy to check different scenarios and see how small changes in the signals and conditions change the behaviour of the model.

Model checking seems to be directly usable for verifying designs of safety I&C systems. Design verification is a key task in the design flow because it can eliminate tricky design errors which are hard to detect later in the development process and are very expensive to repair leading often to a major redesign and reimplementation cycle. An advantage of this approach to more traditional testing and simulation work is that it can provide full coverage of the verification. When model checking system properties, it is often necessary to model the system environment to some degree. Fortunately, modelling languages supported by model checking tools are quite usable for capturing the environment and it is possible to create simple models covering the essential behaviour of the environment.

The modelled arc protection system included timing aspects, especially delays, which seem to be crucial in many safety I&C systems and which are also very challenging to design and verify. For larger and more complicated designs with extensive use of delays and other timing aspects further work is needed to develop robust design and verification techniques.

6 REFERENCES

/1/ B. Berard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, and Ph. Schnoebelen: Systems and Software Verification. Model-Checking Techniques and Tools. Springer, 2001.

/2/ K. Björkman, J. Frits, J. Valkonen, J. Lahtinen, K. Heljanko, I. Niemelä and J .J. Hämäläinen: Verification of Safety Logic Designs by Model Checking, Sixth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009, Knoxville, Tennessee, April 5-9, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL 2009.

/3/ Randal E. Bryant: Symbolic Boolean manipulation with ordered binary decision diagrams. ACM Computing surveys, 24(3):293-318, 1992.

/4/ Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill, and L. J. Hwang: Symbolic model checking: 10^20 states and beyond. Information and Computation, 98(2):142-170, 1992.

/5/ A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, Marco Roveri, R. Sebastiani, and Armando Tacchella: NuSMV 2: An opensource tool for symbolic model checking. In CAV'02, volume 2404 of LNCS, pages 359-364. Springer, 2002.

/6/ Edmund M. Clarke, Orna Grumberg, and Doron A. Peled: Model Checking. The MIT Press, 1999.

/7/ Edmund M. Clarke and E. Allen Emerson: Design and synthesis of synchronization of skeletons using branching time temporal logic. In Proceedings of the IBM Workshop on Logics of Programs, volume 131 of LNCS, pages 52-71. Springer, 1981.

/8/ NuSMV Model Checker v.2.4.3. http://nusmv.irst.itc.it/ 2009.

/9/ J.P. Quielle and J. Sifakis: Specification and verification of concurrent systems in CESAR. In Proceedings of the 5th International Symposium on Programming, pages 337-350, 1981.

/10/ SAFIR2010, The Finnish Research Programme on Nuclear Power Plant Safety 2007 – 2010, http://www.vtt.fi/safir2010, 2009

/11/ UPPAAL integrated tool environment v. 4.0.6, http://www.uppaal.com/ 2009.

/12/ J. Valkonen, V. Pettersson, K. Björkman, J.-E. Holmberg, M. Koskimies, K. Heljanko, and I. Niemelä: Model-Based Analysis of an Arc Protection and an Emergency Cooling System - MODSAFE 2007 Work Report. VTT Working Papers 93, VTT Technical Research Centre of Finland, Espoo, Finland, February 2008, 51 p. http://www.vtt.fi/inf/pdf/workingpapers/2008/W93.pdf

/13/ J. Valkonen, I. Karanta, M. Koskimies, K. Heljanko, I. Niemelä, D. Sheridan, and R. E. Bloomfield: NPP Safety Automation Systems Analysis - State of the Art. VTT Working Papers 94, VTT Technical Research Centre of Finland, Espoo, Finland, February 2008, 63 p. http://www.vtt.fi/inf/pdf/workingpapers/2008/W94.pdf