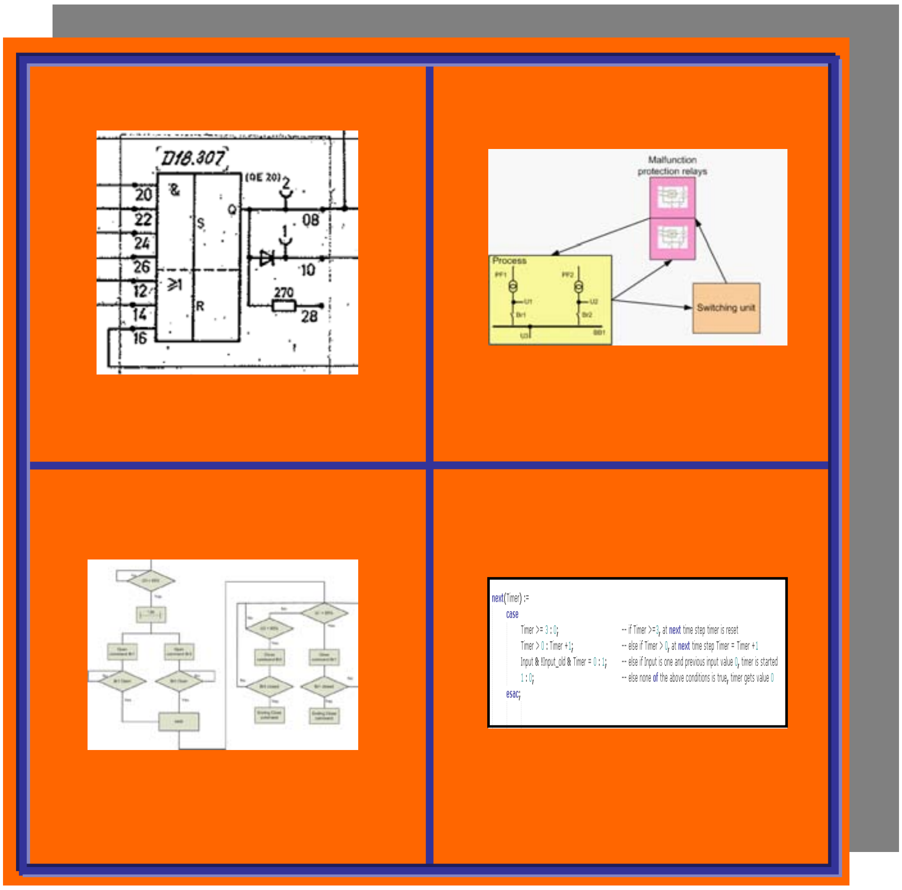
VTT WORKING PAPERS 177



```
next(Timer) :=
    case
        Timer >= 3 : 0;                         -- if Timer >=3, at next time step timer is reset
        Timer > 0 : Timer +1;                   -- else if Timer > 0, at next time step Timer = Timer +1
        Input & !Input_old & Timer = 0 : 1;     -- else if Input is one and previous input value 0, timer is started
        1 : 0;                                  -- else none of the above conditions is true, timer gets value 0
    esac;
```

Kim Björkman, Janne Valkonen & Jukka Ranta

Model-based analysis of an automated changeover switching unit for a busbar

MODSAFE 2009 work report

ISBN 978-951-38-7519-0 (URL: http://www.vtt.fi/publications/index.jsp)
ISSN 1459-7683 (URL: http://www.vtt.fi/publications/index.jsp)

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 3, PL 1000, 02044 VTT
puh. vaihde 020 722 111, faksi 020 722 4374

VTT, Bergsmansvägen 3, PB 1000, 02044 VTT
tel. växel 020 722 111, fax 020 722 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O. Box 1000, FI-02044 VTT, Finland
phone internat. +358 20 722 111, fax +358 20 722 4374

Technical editing Mirjami Pullinen

Series title, number and
report code of publication

VTT Working Papers 177

Author(s)
Kim Björkman, Janne Valkonen & Jukka Ranta

Title

Model-based analysis of an automated changeover switching unit for a busbar

MODSAFE 2009 work report

Abstract

Verification of digital instrumentation and control (I&C) systems is challenging, because programmable logic controllers enable complicated control functions and the state spaces (number of distinct values of inputs, outputs, and internal memory) of the designs become easily too large for comprehensive manual inspection. Model checking is a promising formal method that can be used for verifying the correctness of system designs. A number of efficient model checking systems are available offering analysis tools that are able to determine automatically whether a given state machine model satisfies the desired safety properties. Model checking can also handle delays and other time-related operations, which are crucial in safety I&C systems and challenging to design and verify.

The system analysed in this research project is called "automated changeover switching unit for a busbar" and its purpose is to switch the power feed to stand-by power supply in the event of voltage breaks. The system is modelled as a finite state machine and some of its key properties are verified with the NuSMV model checking tool. The time-dependent components are modelled to operate in discrete fixed-length time steps and the lengths of the timed functions are scaled to avoid state explosion and enable efficient model checking.

ISBN
978-951-38-7519-0 (URL: http://www.vtt.fi/publications/index.jsp)

Series title and ISSN VTT Working Papers 1459-7683 (URL: http://www.vtt.fi/publications/index.jsp)		Project number 32527
Date June 2011	Language English	Pages 23 p.
Name of project MODSAFE – Model-based safety evaluation of automation systems	Commissioned by	
Keywords Model checking, automation system, I&C, NuSMV, system safety, MODSAFE, SAFIR	Publisher VTT Technical Research Centre of Finland P. O. Box 1000, FI-02044 VTT, Finland Phone internat. +358 20 722 4520 Fax +358 20 722 4374	

Preface

This report has been prepared under the research project Model-Based Safety Evaluation of Automation Systems (MODSAFE), which is part of the Finnish Research Programme on Nuclear Power Plant Safety 2007–2010 (SAFIR2010). The goals of the project are to develop methods for model-based safety evaluation, apply the methods in realistic case studies, evaluate the suitability of formal model checking methods for Nuclear Power Plant (NPP) automation analysis, and develop recommendations for the practical application of the methods. This report describes one of the systems investigated during the third project year (2009) of the whole MODSAFE project, introduces its special characteristics and summarizes the results of model checking.

The report is an extended version of an article published in the Proceedings of the 7th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies, NPIC&HMIT 2010[1].

We wish to express our gratitude to the representatives of the organizations who provided us with the case examples and all those who have given their valuable input in the meetings and discussions during the project.

Espoo, September 2010
Authors

[1] Björkman, K., Valkonen, J. & Ranta, J. Verification of Automated Changeover Switching Unit by Model Checking. Proceedings of the 7th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPIC&HMIT 2010), November 7–11, 2010, Las Vegas, Nevada. Pp. 1719–1728.

Contents

1. Introduction

Verification of digital instrumentation and control (I&C) systems is challenging because programmable logic controllers enable complicated control functions and the state spaces (number of distinct values of inputs, outputs, and internal memory) of the designs easily become too large for comprehensive manual inspection. 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, often leading often to a major redesign and reimplementation cycle. Typically, verification and validation (V&V) activities rely heavily on subjective evaluation, which covers only a limited part of the possible behaviours of the system, and therefore more rigorous formal methods are required. Such formal methods have been studied (see, e.g., Valkonen et al. [13] for an overview) but they are not yet widely used.

Model checking [6] is a promising formal method that can be used for verifying the correctness of system designs. Before the Model-based safety evaluation of automation systems (MODSAFE) project, it was not previously applied in the safety evaluation of nuclear power plant (NPP) automation systems (at least in Finland), but internationally it has been used in verifying the correct behaviour of, e.g., hardware and microprocessor designs, data communications protocols and operating system device drivers. A number of efficient model checking systems are available offering analysis tools that are able to automatically determine whether a given state machine model satisfies the desired safety properties. Model checking can also handle delays and other time-related operations, which are crucial in safety I&C systems and challenging to design and verify.

The objective of the MODSAFE project is to evaluate and develop methods based on formal model checking and apply them in the safety analysis of NPP safety automation (I&C). The purpose is to get a group of methods and tools that can support the practical safety evaluation work and benefit utilities, regulators, and vendors. The tasks of the first two project years included reviewing the state of the art of employing formal methods and models for safety evaluation of industrial and nuclear safety systems [13], developing basic methodology for applying model checking to safety evaluation, and studying the feasibility of the approach [3, 4, 8, 14, 15, 16]. In general, the project focuses on a number of case studies which direct the development of the required methodology and serve as benchmarks for evaluating the feasibility and applicability of the approach.

This report summarizes the experiences gained during the project year 2009 while working on a case study called “Automated changeover switching unit for a busbar”. The system has a different nature compared to the systems analysed earlier in the project because there are analogue features and a wide range of different timed functions.

Section 1 is the introduction. Section 2 provides some background information on model checking. Section 3 presents the automated changeover switching unit for a busbar. Section 4 explains how the system was modelled and introduces the results of model checking and system analysis. Section 5 concludes the report.

2. Model checking

Model checking [6, 7, 11] is a computer-aided verification method developed to formally verify the correct functioning of a system design model by examining all of its possible behaviours. The models used in model checking are quite similar to those used in simulation, as basically the model must describe the behaviour of the system design for all sequences of inputs. However, unlike simulation, model checkers examine the behaviour of the system design with all input sequences and compare it with the system specification. In model checking, at least in principle, the analysis can be fully automated with computer-aided tools. The specification is expressed in a suitable specification language, temporal logics being a prime example, describing the allowed behaviours of a system. Given a model and a specification as inputs, a model checking algorithm 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 counter-example execution of the system demonstrating why the property is violated.

The MODSAFE project has been using two model checkers, NuSMV [5, 10] – which was originally designed for hardware model checking – and UPPAAL [12], which supports model checking of timed automata. In the analysis of the case described in this report, the NuSMV model checker was used. It is briefly introduced below.

NuSMV [5, 10] is a state-of-the-art symbolic model checker that supports synchronous state machine models where the real-time behaviour has to be modelled with discrete time steps using explicit counter variables that are incremented at a common clock frequency. NuSMV supports model checking using both Linear Temporal Logic (LTL) and Computation Tree Logic (CTL) [6], making it quite flexible in expressing design specifications. The model checking algorithms employed in this work are based on symbolically representing and exploring the state space of the system by using Binary Decision Diagrams (BDDs) [3, 9]. In addition, SAT (Propositional Satisfiability)-based bounded model checking [1] is also supported by NuSMV [2] for finding bugs in larger designs. The sophisticated model checking techniques used by NuSMV can handle non-determinism induced by free input variables well, but modelling the real-time aspects can be more challenging due to the inherently discrete time nature of the synchronous state machine model employed by NuSMV.

3. Description of the automated changeover switching unit for a busbar

The system modelled and analysed in the research is called “automated changeover switching unit for a busbar”. The purpose of the system is to switch the power feed to alternative power supply when there is a voltage break that lasts over 1 s. The distribution network consists of several busbars of different voltages supplying power to, e.g. control systems and motor operated valves. The process that is controlled by the modelled switching unit is presented in Figure 1.

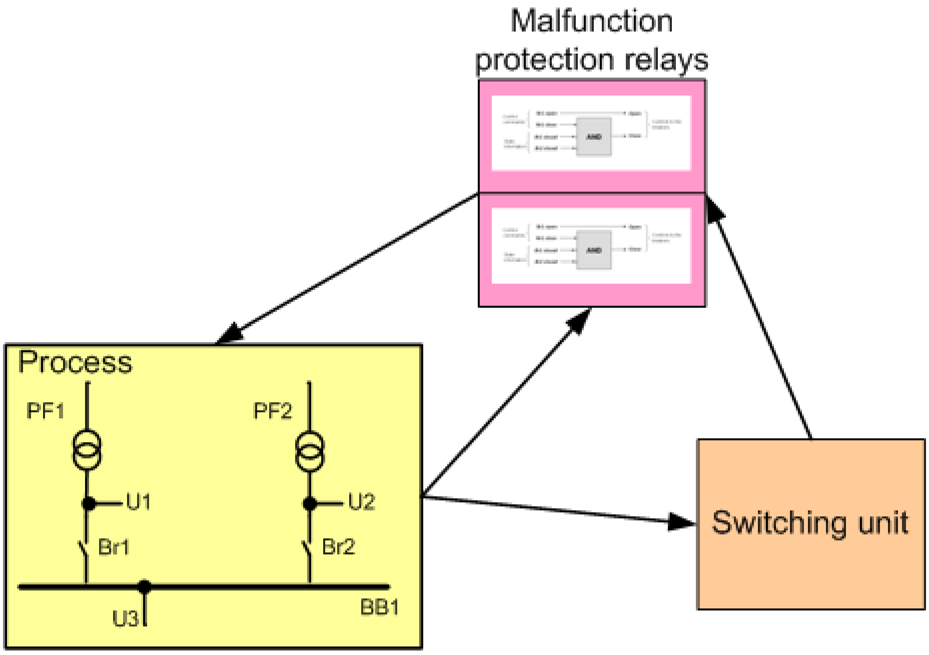


Figure 1. The overall system.

The modelled switching unit is a backup system for the automated rapid changeover switching unit and is implemented using traditional analogue technology, whereas the rapid switching unit is based on digital technology. The two changeover switching units are completely independent of each other. Aside from these automated switching units, manual actuation is also possible from the control room.

The overall system consists of three separate parts (see Figure 1):

1. The process surrounding the switching unit (the system environment).
2. The switching unit (control logic).
3. The malfunction protection relays.

The process surrounding the changeover switching unit consists of two alternative power feeds PF1 (primary) and PF2 (secondary) that can feed power to the busbar BB1. The changeover switching unit controls the feed breakers Br1 and Br2. In normal power plant operation, the switching unit is passive. In the event that the voltage of BB1 is below 65% of nominal-voltage (U3 < 65%) for 1.6 s, the switching unit is triggered.

Figure 2 illustrates a sequence chart describing the functionality of the switching unit. In the event of voltage loss (U3 < 65%), both feed breakers are given the "open" command. When both breakers are open, the breaker Br1 is given the "close" command if the voltage of PF1 is higher than 85% of the nominal-voltage (U1 > 85%). If U1 is below 85% and the voltage of PF2 is higher than 85% of the nominal-voltage (U2 > 85%), Br2 is given the "close" command. If the voltage level of both power feeds is below 85%, the switching unit waits until one of the power feeds has the required voltage level and the "close" command is given to the respective breaker. If the voltage returns to both power feeds simultaneously, Br1 is prioritised.

3. Description of the automated changeover switching unit for a busbar

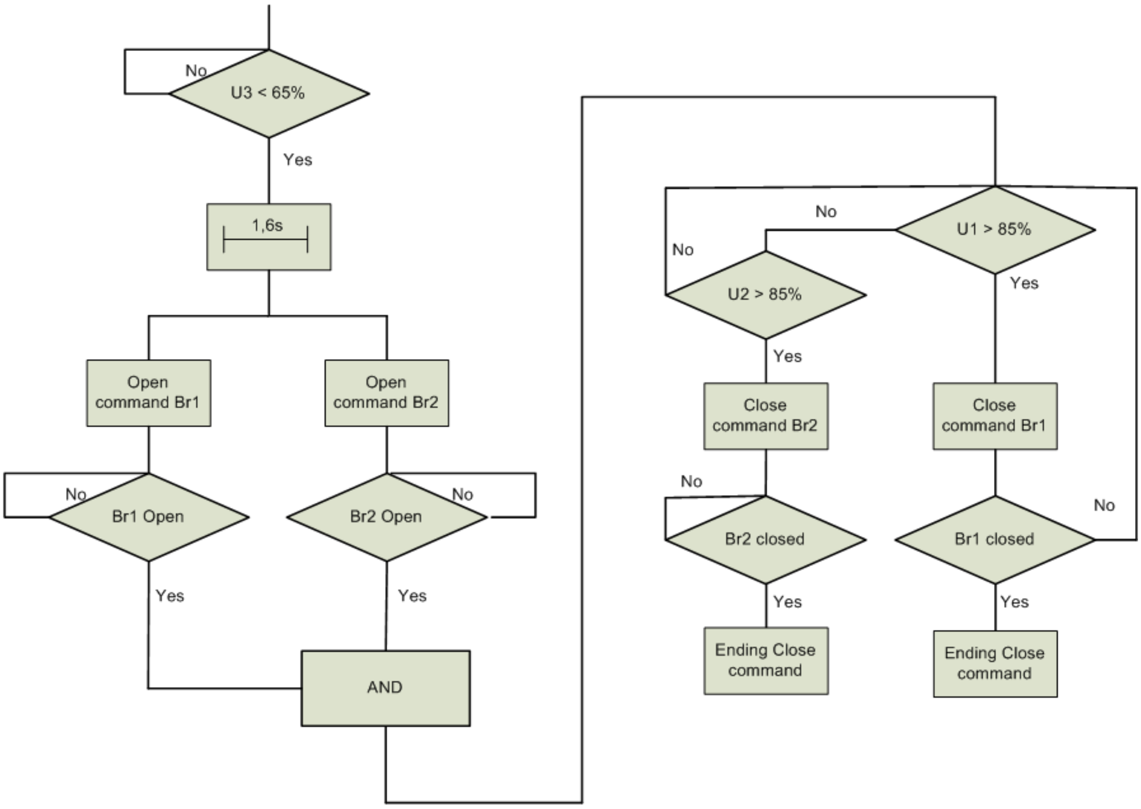


Figure 2. Sequence chart outlining the functionality of the switching unit.

The control commands of the logic are first transmitted to the malfunction protection relays and from there to the breakers. The functionality of the relay Br1 is presented in Figure 3 (identical to the relay Br2). The relay has a close-blocking function, that is, if one of the breakers is closed, further close commands are blocked by the relay.

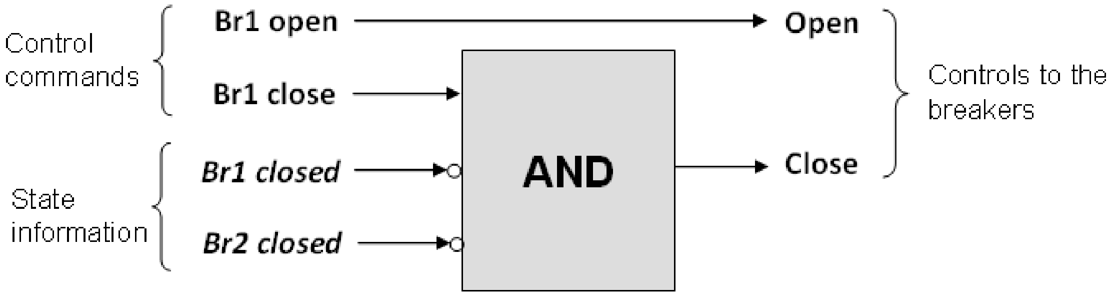


Figure 3. Functionality of the relay Br1.

4. Modelling the automated changeover switching unit for a busbar

4.1 Model of the system

The formal models created for analysing the automated changeover switching unit are based on the logic diagrams in the system documentation and discussions as well as the comments of the system experts. The created NuSMV model comprises of a set of modules that are collections of declarations, constraints, and specifications. The functional entities of the system were separated into a variety of modules. Since a module can contain instances of other modules, a structural hierarchy was constructed. Figure 4 below illustrates the modules and their hierarchy.

The module “Main” is the highest in the hierarchy of the system model. The instances of the other modules “Switching unit”, “Malfunction protection relay” and “Breakers” are created there. The modules “Main” and “Breakers” together constitute the model of the environment surrounding the switching unit. The “Main” module includes the process inputs and behaviour of the busbar and the power feeds. The “Main” module also contains properties and conditions that were model-checked. The “Malfunction protection relay” module was modelled as a black box that functions as realistically as possible. The relay had an arbitrary operation cycle between 700 ms and 1200 ms which was implemented using a counter variable that ranged from 0 to 1200. The counter increases by one at each state transition until it reaches the value 700. Thereafter, the behaviour of the counter changes to nondeterministic, meaning that there are two options at each state transition: either the counter value was increased by one or it was given the value 1200. After the final value of 1200 is reached, the counter is reset and the cycle starts anew. At the counter value 0, inputs of the malfunction protection relay are read, and at the final value 1200 the outputs are updated. Figure 5 illustrates the implementation of the counter.

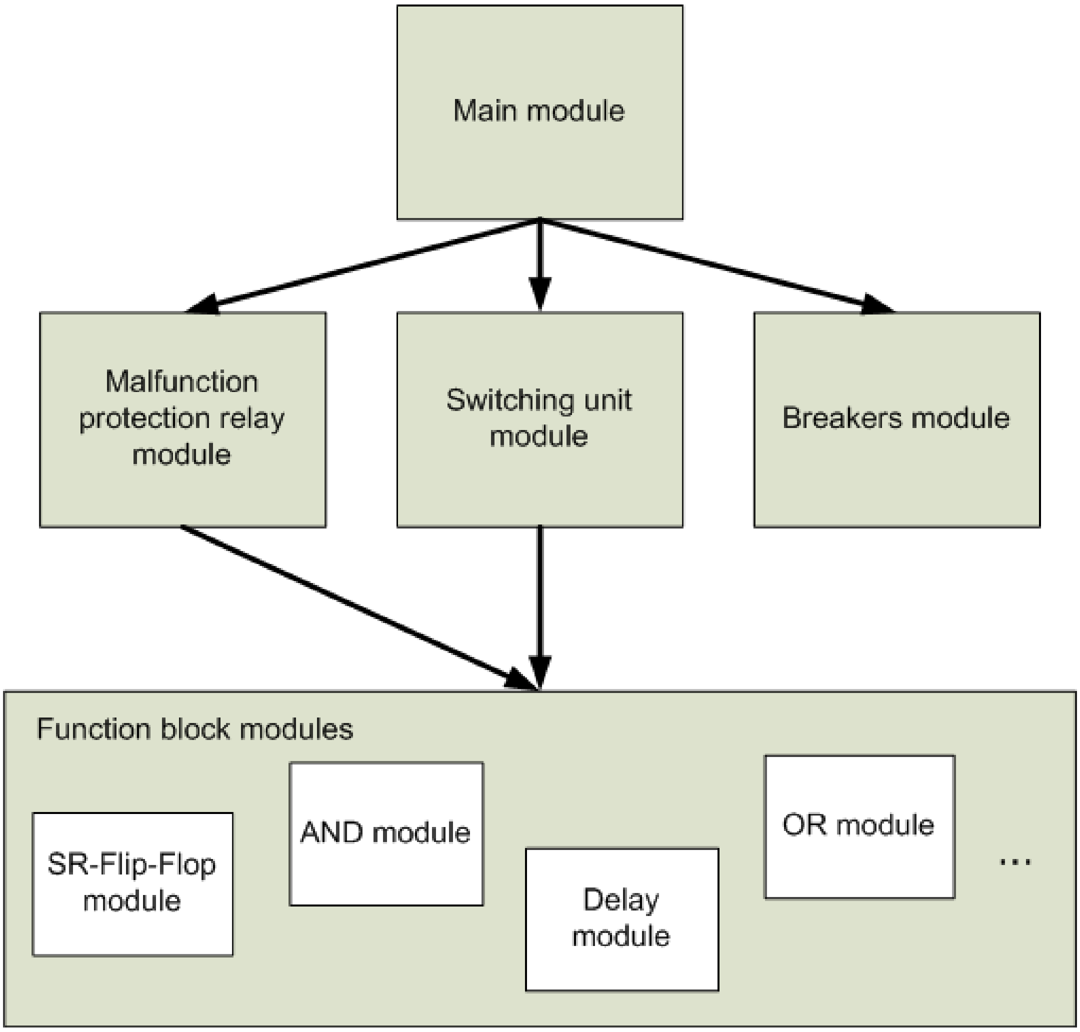
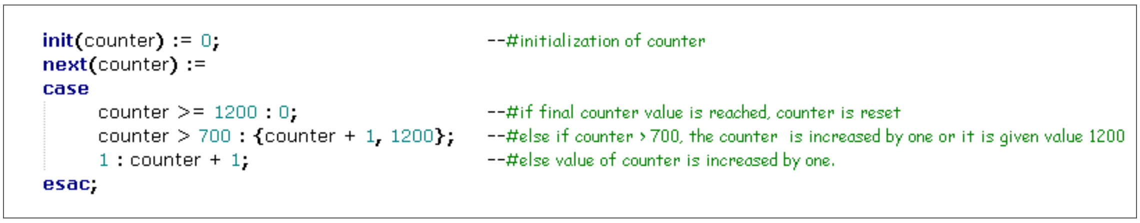


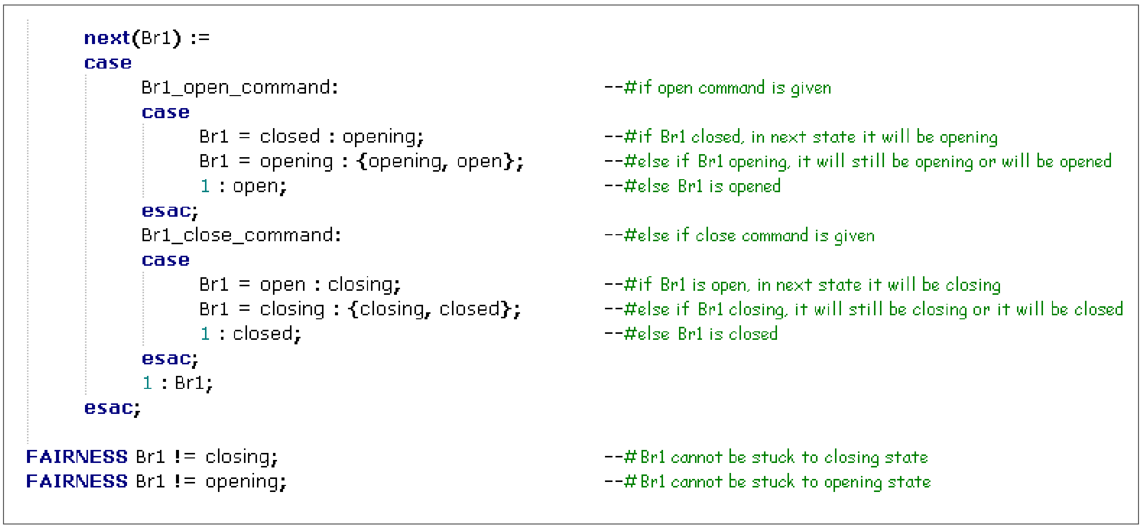
Figure 4. Module hierarchy.



```
init(counter) := 0;                           --#initialization of counter
next(counter) :=
case
        counter >= 1200 : 0;                  --#if final counter value is reached, counter is reset
        counter > 700 : {counter + 1, 1200};  --#else if counter > 700, the counter  is increased by one or it is given value 1200
        1 : counter + 1;                      --#else value of counter is increased by one.
esac;
```

Figure 5. Counter implementation of malfunction protection relay.

In the “Breakers” module, each breaker was modelled as a variable with four states {open, opening, closing, closed}. When a breaker is in the state “open” and receives a “close” command, it first changes its state to “closing”. After that, it can remain arbitrarily in the “closing” state before it finally changes to the “closed” state (similarly for “open” command) (see Figure 6 for breaker Br1). By allowing the breaker to stay at the “closing” state for an arbitrary time, the possible delays in the breaker’s operation can be included in the model.



```
        next(Br1) :=
        case
                Br1_open_command:                               --#if open command is given
                case
                        Br1 = closed : opening;                 --#if Br1 closed, in next state it will be opening
                        Br1 = opening : {opening, open};        --#else if Br1 opening, it will still be opening or will be opened
                        1 : open;                               --#else Br1 is opened
                esac;
                Br1_close_command:                              --#else if close command is given
                case
                        Br1 = open : closing;                   --#if Br1 is open, in next state it will be closing
                        Br1 = closing : {closing, closed};      --#else if Br1 closing, it will still be closing or it will be closed
                        1 : closed;                             --#else Br1 is closed
                esac;
                1 : Br1;
        esac;

FAIRNESS Br1 != closing;                                        --#Br1 cannot be stuck to closing state
FAIRNESS Br1 != opening;                                        --#Br1 cannot be stuck to opening state
```

Figure 6. Implementation of a breaker’s behaviour.

The basic function blocks were modelled as individual modules to improve component reusability. Both the “switching unit” and the “malfunction protection relay” modules utilize the function blocks to realize their functionality.

4.2 Assumptions and restrictions

When modelling the system, it was assumed that the system was set to automatic control mode. Thus, no manual actions were considered. The rapid changeover switching unit was assumed to be unavailable because the analysis was focused on the functionality of the automated switching unit. The unavailability of the rapid changeover switching unit could be due to, e.g. maintenance or a malfunction. Assumptions regarding the timed functions are discussed in the following section.

4.3 Managing timed functions

As mentioned earlier, the system consists of several timed functions with different time scales:

- 1 and 8 ms delays caused by the inner operation of some logical blocks
- logical blocks with delays of 45–110 ms
- 1.6 s delay blocks

- 2 s time pulse blocks
- malfunction protection relays with an operational cycle between 0.7–1.2 s.

Because NuSMV does not support continuous time, the time-dependent components were modelled to operate in discrete fixed-length time steps. The first version of the model (original model) included all the original delays without scaling. However, model checking was not feasible, and the delays had to be scaled based on the following reasons:

1. Excessively long computation time. If the timed functions were kept as they were and all delays were included in the model, model checking would have been unfeasible because of impractically lengthy computation times.
2. Overly complicated counter-examples. The possible counter-examples due to violated properties could contain so many state transitions that their manual analysis would be too arduous a task within the scope of this research.

The objective of scaling was to retain all relevant system behaviour in the model while avoiding the state explosion problem. Long-lasting (over 500 ms) timed functions were considered as the most relevant for the system behaviour. The aim was to keep the interrelationship of the lengths of the long-lasting timed functions unaltered. Short-term delays could not be scaled with the same coefficient, because the delays below 40 ms would have been completely omitted and the behaviour of the model in some borderline cases would have been lost. Therefore the scaling was not done linearly. The timed functions were scaled with a different coefficient selected with consideration for the ratio of the long and short delays. The short delays had to sufficiently fit many times inside the long delays to enable the model to behave as realistically as possible, i.e. to capture all relevant features of the system. The scaling of the delays is presented in Table 1, and Figure 7 illustrates the relation between the original and the scaled delays.

Table 1. Original and scaled delays.

Original delays (ms)	Scaled delays (ms)
1	0
8	1
45	4
110	10
700	40
1200	70
1600	90
2000	110

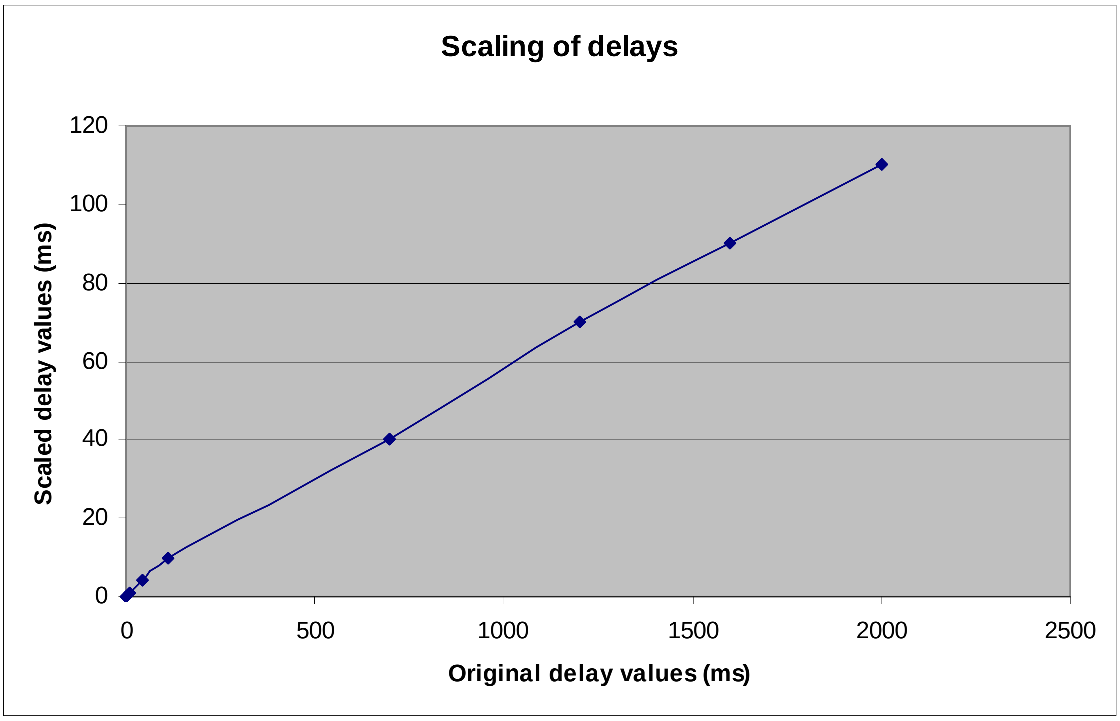


Figure 7. Scaling of delays.

Because the time behaviour in the scaled model is much rougher than in the original model, it can be said with considerable certainty that all the behaviours of the scaled model are included in the analyzed system. However, all the behaviours of the analyzed system may not be included in the scaled model, because some of the shortest delays are omitted and the scaling is not linear.

The discretized model of the system contained feedback loops in which the output of a function block affects the input of the same function block during the same time step. Therefore, some additional unit delay blocks were used in the model to eliminate such loops.

The NuSMV model was checked against the following properties:

1. If the voltage level of BB1 is below 65% (U3 < 65%) and in either PF1 or PF2 the voltage level is over 85% (U1 > 85% or U2 > 85%) (can alter at each time step), then eventually the voltage level of BB1 is no longer below 65%.
2. The breakers can never be closed at the same time.
3. The breakers cannot receive close and open commands at the same time.
4. If the voltage level of BB1 is over 65%, no actions are taken.

5. If the voltage level is below 65% and in either PF1 or PF2 the voltage level is over 85% (one of the power feeds is continuously powered), then eventually the voltage level of BB1 is no longer below 65%.

In property 1, the power feed powered can alter at each time step, whereas in property 5, one of the power feeds is continuously powered. Property 5 is used for analysing tolerance of disturbances.

4.4 Results

Two possible errors were found during the analysis of the system model. Property 1 is violated if a powerless power feed is connected to the busbar, in which case the breakers cannot be reopened. Figure 8 illustrates a part of the control logic that concerns the opening of the breakers. A powerless power feed could be connected to the busbar if, e.g. the respective power feed loses its power during the closing of the breaker. If a powerless power feed is connected to the busbar, the breakers cannot be reopened, because a “close” command combined with respective closed breaker generates a signal that sets a set-reset flip-flop (SR1), which in turn resets the set-reset flip-flop (SR2) controlling the opening sequence (as well as flip-flops controlling the closing sequences). The set-reset flip-flop SR1 is reset by the signal indicating a high voltage level in busbar BB1. Additionally, setting SR2 requires that the output of the pulse block (PB1) is 1, which is basically triggered by a rising edge of the signal indicating low voltage in BB1. Because the low voltage signal is continuously 1, PB1 cannot be retriggered.

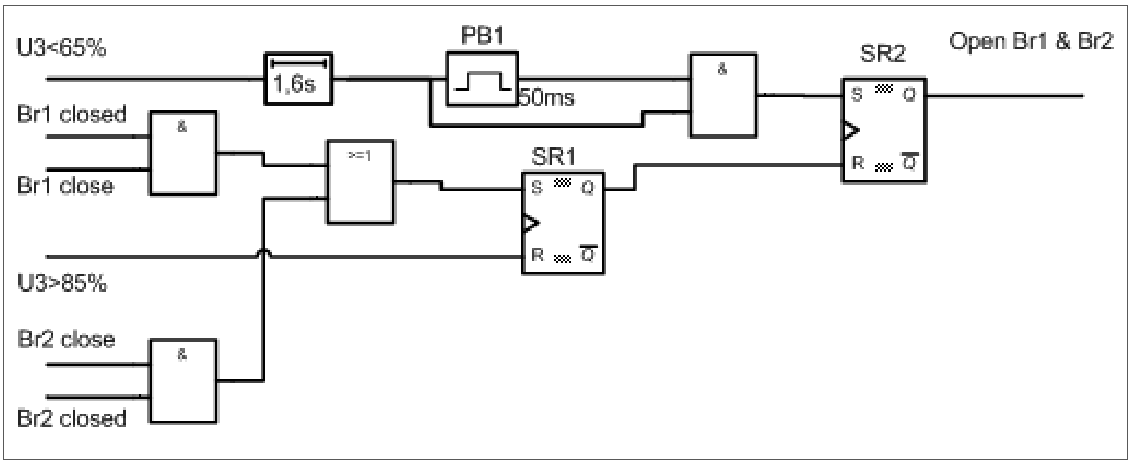


Figure 8. Part of the switching unit logic controlling the opening of the breakers.

Property 2 was violated in a case where the closing and opening of the breakers could last for an arbitrary time. First, both breakers are opened, and then Br1 receives a close command. If the closing of the breaker takes over 2 s (the breaker is given a close command for 2 s controlled by a 2 s time pulse block), and the voltage of PF1 is lost while the breaker is closing, and if PF2 has the required voltage, it will receive a “close” command after the 2 s delay. Because the original “close” command of Br1 cannot be cancelled, both of the breakers can simultaneously be closed. If the opening and closing of the breakers is limited to a few state transitions (under 2 s), the property is not violated. The design was found to operate correctly with respect to properties 3 and 4.

Property 5 was used for analysing the tolerance of disturbances allowed to occur in the contacts of the voltage measurements of the power feeds. Due to the disturbance, the voltage level measurement signal could indicate low voltage, even though in reality the voltage of the power feed is high. The modified property 1 holds if the disturbance is momentary, whereas the property is violated if the disturbance is continuous.

The model checking was carried out with NuSMV version 2.4.3 on a PC with 16GB of RAM and an Intel Xeon CPU X5460 processor running at 3.16GHz. The model checking times were between 2 s and 18h depending on the checked property. The total number of states in the model was $2 \cdot 10^{28}$, out of which $2 \cdot 10^{9}$ were reachable. The time step utilised was 2 ms. When using a 1 ms time step, the model-checking times were unreasonably long.

5. Conclusions

Verification of I&C systems is a challenging task, especially from the safety perspective. Modern digitalized I&C systems enable complicated control tasks, and due to system modernizations a single application can contain both analogue and digital parts. This makes designing and verification of such systems complicated.

Model checking is a promising formal method that enables complete verification of designs of such systems. Model checking requires a state machine model of the design and its relevant environment. It seems to lend itself well to verification of safety logic designs, because formulating a state machine model of such a design is often quite unproblematic.

In the MODSAFE project, the use of model checking was studied to verify a control logic combining analogue and digital technology. The analysis was done using NuSMV model checking tool, which employs finite state machines typically used for verifying hardware.

The analysed system contained several timing functions of different lengths varying in the range of 1 ms to 2 s. NuSMV does not support continuous time, thus time dependent functions had to be modelled to operate in discrete time with fixed length steps. To make model checking feasible with NuSMV, the lengths of the timing functions had to be decreased by scaling them down. Long-lasting functions were scaled with a different coefficient than the short-time functions. The goal of the scaling was to keep all relevant behaviour of the system intact. It is possible that the scaling may influence the completeness of the verification, thus some relevant behaviour of the system may not be included in the model. However, in this case it was concluded that the shortest scaled delays sufficiently fit in the long delays many times to enable the model to behave as realistically as possible. It was also concluded that ignoring the shortest delay of the real system should not influence the results of the model checked properties. The scaled model may contain some behaviours that are impossible in the real system, which is not problem because the possible unrealistic behaviour can be eliminated by examining counter-examples and modifying the model.

The same analysis was also performed with linear scaling and the model checking results were similar to the non-linear case. The non-linear scaling was considered better because in linear scaling the two shortest delays would have been omitted compared to omitting only the shortest delay in the non-linear model. In both scaling scenarios there is no common factor for all the delays i.e. in any case some scaled delays must be rounded to the closest integer.

The analysis showed how challenging model checking timed systems is and that the limits of the NuSMV model checking tool can be easily reached when the model contains several timed functions of various lengths. This brings out the need for simplification of the model and scaling of the timed functions.

It was already realised in the previous case studies that NuSMV performs well when the large state space of the model is caused by a high number of input variables. This case study was the first one where the state explosion problem was caused by the timed functions and solved with scaling. Based on the experiences of this and the previous cases studies, the Uppaal model checker could be more suitable than NuSMV in these kinds of verification tasks. However, it was valuable and interesting to see how NuSMV performs and how the scaling of the timed functions affects the model. An interesting topic for further research could be to implement the same system with an Uppaal model checking tool and compare the results and performances of the two tools.

References

1. Biere, A., Cimatti, A., Clarke, E. M. & Zhu, Y. 1999. Symbolic model checking without BDDs. In: Proc. of the Fifth International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'99). In Proceedings of the Fifth International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS '99). Amsterdam, the Netherlands. Lecture Notes in Computer Science, Vol. 1579, Springer, 1999. Pp. 193–207.

2. Biere, K., Heljanko, T., Junttila, T., Latvala & Schuppan, V. 2006. Linear Encodings of Bounded LTL Model Checking. Logical Methods in Computer Science 2(5:5), pp. 1–64.

3. Björkman, K., Frits, J., Valkonen, J., Lahtinen, J., Heljanko, K., Niemelä, I. & Hämäläinen, J. J. 2009. Verification of safety logic designs by model checking. In: Proceedings of the Sixth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT 2009, Knoxville, Tennessee, April 2009.

4. Björkman, K., Frits, J., Valkonen, J., Heljanko, K. & Niemelä, I. 2009. Model-based analysis of a stepwise shutdown logic. MODSAFE 2008 Work Report. Espoo: VTT Working Papers 115. 36 p. + app. 4 p. http://www.vtt.fi/inf/pdf/workingpapers/2009/W115.pdf

5. Cavada, R., Cimatti, A., Jochim, C. A., Keighren, G., Olivetti, E., Pistore, M., Roveri, M. & Tchaltsev, A. 2005. NuSMV 2.4 User Manual. CMU and ITC-irst.

6. Clarke, E. M., Grumberg, O. & Peled, D. A. 1999. Model Checking. The MIT Press.

7. Clarke, E. M. & Emerson, E. A. 1981. 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. Springer. Pp. 52–71.

8. Lahtinen, J. 2008. Model checking timed safety instrumented systems. Vol. 3. Espoo: Helsinki University of Technology. TKK reports in information and computer science. ISBN 978-951-22-9445-9. http://lib.tkk.fi/Reports/2008/isbn9789512294459.pdf.

9. McMillan, K. L. 1993. Symbolic Model Checking. Kluwer Academic Publ.

10. NuSMV Model Checker v.2.4.3, 2008. http://nusmv.irst.itc.it/.

11. Quielle, J. & Sifakis, J. 1981. Specification and verification of concurrent systems in CESAR. In: Proceedings of the 5th International Symposium on Programming. Pp. 337–350.

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

13. Valkonen, J., Karanta, I., Koskimies, M., Heljanko, K., Niemelä, I., Sheridan, D. & Bloomfield, R. E. 2008. NPP Safety Automation Systems Analysis. State of the Art. Espoo: VTT Working Papers 94. 62 p. http://www.vtt.fi/inf/pdf/workingpapers/2008/W94.pdf.

14. Valkonen, J., Pettersson, V., Björkman, K., Holmberg, J.-E., Koskimies, M., Heljanko, K. & Niemelä, I. 2008. Model-Based Analysis of an Arc Protection and an Emergency Cooling System. MODSAFE 2007 Work Report. Espoo: VTT Working Papers 93. 13 p. + app. 38 p. http://www.vtt.fi/inf/pdf/workingpapers/2008/W93.pdf.

15. Valkonen, J., Koskimies, M., Pettersson, V., Heljanko, K., Holmberg, J.-E., Niemelä, I. & Hämäläinen, J. J. 2008. Formal Verification of Safety I&C System Designs: Two Nuclear Power Plant Related Applications. Enlarged Halden Programme Group Meeting. Proc. Man-Technology-Organisation Session. Loen, Norway, 18–23 May.

16. Valkonen, J., Koskimies, M., Björkman, K., Heljanko, K., Niemelä, I. & Hämäläinen, J. J. 2009. Formal verification of safety automation logic designs. In: Automaatio XVIII 2009 Seminaari.

VTT Working Papers

138 Tapio Salonen, Juha Sääski, Charles Woodward, Mika Hakkarainen, Otto Korkalo & Kari Rainio. Augmented Assembly – Ohjaava kokoonpano. Loppuraportti. 2009. 32 s. + liitt. 36 s.

139 Jukka Hietaniemi & Esko Mikkola. Design Fires for Fire Safety Engineering. 2010. 100 p.

140 Juhani Hirvonen, Eija Kaasinen, Ville Kotovirta, Jussi Lahtinen, Leena Norros, Leena Salo, Mika Timonen, Teemu Tommila, Janne Valkonen, Mark van Gils & Olli Ventä. Intelligence engineering framework. 2010. 44 p. + app. 4 p.

141 Juha Forström, Esa Pursiheimo, Veikko Kekkonen & Juha Honkatukia. Ydinvoimahankkeiden periaatepäätökseen liittyvät energia- ja kansantaloudelliset selvitykset. 2010. 82 s. + liitt. 29 s.

142 Ulf Lindqvist, Maiju Aikala, Maija Federley, Liisa Hakola, Aino Mensonen, Pertti Moilanen, Anna Viljakainen & Mikko Laukkanen. Hybrid Media in Packaging. Printelligence. 2010. 52 p. + app. 7 p.

143 Olavi Lehtoranta. Knowledge flows from incumbent firms to newcomers. The growth performance of innovative SMEs and services start-ups. 2010. 36 p. + app. 2 p.

144 Katri Grenman. The future of printed school books. 2010. 42 p.

145 Anders Stenberg & Hannele Holttinen. Tuulivoiman tuotantotilastot. Vuosiraportti 2009. 2010. 47 s. + liitt. 5 s.

146 Antti Nurmi, Tuula Hakkarainen & Ari Kevarinmäki. Palosuojattujen puurakenteiden pitkäaikaistoimivuus. 2010. 39 s. + liitt. 6 s.

147 Juhan Viitaniemi, Susanna Aromaa, Simo-Pekka Leino, Sauli Kiviranta & Kaj Helin. Integration of User-Centred Design and Product Development Process within a Virtual Environment. Practical case KVALIVE. 2010. 39 p.

149 Tommi Ekholm. Achieving cost efficiency with the 30% greenhouse gas emission reduction target of the EU. 2010. 21 p.

150 Sampo Soimakallio, Mikko Hongisto, Kati Koponen, Laura Sokka, Kaisa Manninen, Riina Antikainen, Karri Pasanen, Taija Sinkko & Rabbe Thun. EU:n uusiutuvien energialähteiden edistämisdirektiivin kestävyyskriteeristö. Näkemyksiä määritelmistä ja kestävyyden todentamisesta. 130 s. + liitt. 7 s.

151 Ian Baring-Gould, Lars Tallhaug, Göran Ronsten, Robert Horbaty, René Cattin, Timo Laakso, Michael Durstewitz, Antoine Lacroix, Esa Peltola & Tomas Wallenius. Wind energy projects in cold climates. 2010. 62 p.

152 Timo Laakso, Ian Baring-Gould, Michael Durstewitz, Robert Horbaty, Antoine Lacroix, Esa Peltola, Göran Ronsten, Lars Tallhaug & Tomas Wallenius. State-of-the-art of wind energy in cold climates. 2010. 69 p.

ISBN 978-951-38-7519-0 (URL: http://www.vtt.fi/publications/index.jsp)
ISSN 1459-7683 (URL: http://www.vtt.fi/publications/index.jsp)