Action-based verification of RTCP-nets with CADP

Jerzy Biernacki, Agnieszka Biernacka and Marcin Szpyrka

AGH University of Science and Technology, Department of Applied Computer Science
Al. Mickiewicza 30, 30-059 Krakow, Poland, `{jbiernac, abiernac, mszpyrka}@agh.edu.pl`

Abstract. The paper presents an RTCP-nets' (real-time coloured Petri nets) coverability graphs into Aldebaran format translation algorithm. The approach provides the possibility of automatic RTCP-nets verification using model checking techniques provided by the CADP toolbox. An actual fire alarm control panel system has been modelled and several of its crucial properties have been verified to demonstrate the usability of the approach.

Keywords: RTCP-nets, Petri nets, formal models, CADP, μ calculus, verification, model checking
PACS: 07.05.Tp

INTRODUCTION

System verification is used to establish that the design or product under consideration possesses certain properties. Model checking is an automated technique that, given a finite-state model of a system and a formal property, checks whether this property holds or not for that model [1]. The verification process may be oriented towards states or actions i.e. the formal properties may describe some properties of reachable states or properties of sequences of performed actions. Moreover, there are approaches based on the Petri net structure i.e. net decomposition [2], [3]. In the presented approach the model takes the form of an RTCP-net [4], [5] i.e. an adjust class of timed coloured Petri nets (CP-nets) [6], [7] designed to model real-time systems, especially embedded control systems. The considered verification process is oriented towards actions. We use the CADP toolbox [8] to check whether the model satisfies requirements given as regular alternation-free μ-calculus formulae [9], [10], [11]. The presented action-based approach completes the state-based approach presented in [12]. To reach the aim an algorithm of translation of an RTCP-net coverability graph into Aldebaran format has been formulated and implemented. The paper contains the translation algorithm together with an RTCP-net example used to illustrate usefulness of the approach.

COVERABILITY GRAPH TO ALDEBARAN TRANSLATION ALGORITHM

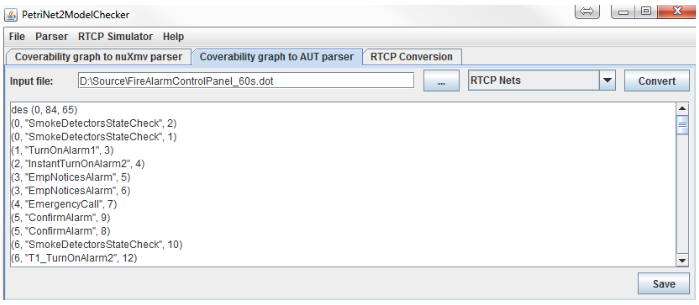
The set of reachable states $\mathscr{R}(M_0,S_0)$ of an RTCP-net is represented using so-called *coverability graph*, which is a labelled transition system with nodes corresponding to states and arcs corresponding to system actions. A state is a pair (M,S), where M is the vector of places' markings and S is the vector of clocks' values. The fact that firing a transition t in a binding b leads from state (M_1,S_1) to (M_2,S_2) is represented by an arc which connects corresponding states [4]. This paper presents action-based approach to RTCP-nets verification. Information about states of the considered model is ignored and the focus is put on the edges of the coverability graph. Graph translated to Aldebaran format preserves its number of nodes and order numbers of the states. It also retains transitions' labels.

The translation algorithm of an RTCP-net coverability graph into the Aldebaran format is presented in Figure 1. The first line initializes the set of defined labels for reachable states. It is denoted with letter L. Lines 2–5 contain label creation for each state (M,S). The label of a state is its order number. Line 9 appends a template content to the output file. Keywords of the template are replaced with appropriate values defined in lines 6–8. Lines 11–15 are processed for each edge of the coverability graph. (M_i,S_i) corresponds to the input node and (M_j,S_j) corresponds to the output node of the edge. Line 15 appends a template that contains information about the transition which firing is denoted by the given edge. As previously, its keywords are then replaced with appropriate values (lines 12–14).

The presented algorithm has been implemented in the PetriNet2ModelChecker tool. This application allows automatic translation of coverability graphs of RTCP, CP and PT (place-transitions) nets into CADP Evaluator input files in the Aldebaran format. The aim of the translation is to enable Petri net validation with μ-calculus formulae. The tool was written in Java language. Swing library was used to provide intuitive graphical user interface (Figure 2). Due

```
$L \leftarrow \emptyset$                                        ▷ the set of defined labels for reachable states
for $i = 0 \ldots |\mathcal{R}(M_0, S_0)|$ do
    for $(M_i, S_i)$ create label $i$
    add label $i$ to $L$
end for
<initial-state> $\leftarrow L((M_0, S_0))$
<number-of-transitions> $\leftarrow |A|$
<number-of-states> $\leftarrow |\mathcal{R}(M_0, S_0)|$
append line des(<initial-state>, <number-of-transitions>, <number-of-states>)
for all $(M_i, S_i) \in \mathcal{R}(M_0, S_0)$ do
    for all $(M_j, S_j)\colon \exists(b, t) \in \mathcal{B}(M_i, S_i) \xrightarrow{(t,b)} (M_j, S_j)$ do
        <from-state> $\leftarrow L((M_i, S_i))$
        <label> $\leftarrow t$
        <to-state> $\leftarrow L((M_j, S_j))$
        append line (<from-state>, <label>, <to-state>)
    end for
end for
```

Figure 1. Coverability graph to Aldebaran translation algorithm.



Input file: D:\Source\FireAlarmControlPanel_60s.dot

RTCP Nets

```
des (0, 84, 65)
(0, "SmokeDetectorsStateCheck", 2)
(0, "SmokeDetectorsStateCheck", 1)
(1, "TurnOnAlarm1", 3)
(2, "InstantTurnOnAlarm2", 4)
(3, "EmpNoticesAlarm", 5)
(3, "EmpNoticesAlarm", 6)
(4, "EmergencyCall", 7)
(5, "ConfirmAlarm", 9)
(5, "ConfirmAlarm", 8)
(6, "SmokeDetectorsStateCheck", 10)
(6, "T1_TurnOnAlarm2", 12)
```

Figure 2. PetriNet2ModelChecker tool.

to the integration with the RTCP-net compiler, the application enables RTCP-net model import, coverability graph generation and translation to Aldebaran format in a few simple steps. PetriNet2ModelChecker also incorporates implemented algorithms of translation coverability graphs into nuXmv language [12], [13]. More details can be found on `http://fm.kis.agh.edu.pl` website.

CASE STUDY

A common practise in construction of fire alarm control panels [14], aiming at the reduction of false fire alarms, is alarm variants usage. The most popular variant is the two-stage alarming which scheme is presented in Figure 3. This approach requires personnel participation who have strictly defined role of the operator in alarm verification process. A model of this system was created with the RTCP-net formalism. The designed net is shown in Figure 4.

A current state of the panel is defined by the color of the token in place `FACP`. There are four possible states of the system: normal, internal alarm, external alarm and terminal. Smoke detection by one of smoke detectors (which corresponds to firing of `SmokeDetectorsStateCheck` transition with `defState` variable set to `warning`) raises the internal alarm (`TurnOnAlarm1`). The internal alarm calls in the operator to identify the fire hazard. The central system determines the time `T1` for operator to confirm reception of the notification. In the presented model `T1` is set to 60 time units. `Clock1` place performs a role of the timer which activates external alarm at the expiry of the deadline (`T1_TurnOnAlarm2`). The external alarm results in launching of fire emergency procedures, including calling the fire department. An acknowledgement of the internal alarm (`ConfirmAlarm`) results in activation of a second timer (`Clock2`). Operator has `T2` time units to assess the threat and verify the alarm. Upon recognition of fire employee has two options. The first one is to push manual call point button (`Emp_TurnOnAlarm2`) which automatically turns on the external alarm. The second option is an attempt to extinguish the fire using available fire fighting equipment. After getting the situation under control personnel has to turn off the internal alarm (`TurnOffAlarm1`). Otherwise, after expiration of the time limit T2 the external alarm is raised (T2_TurnOnAlarm2). T2 is set to 180 time units. Provided that raising the external alarm causes serious consequences, e.g. stoppage of technological processes or activation of automatic extinguishing system, coincidence detection is often used. It is one of the most effective ways of elimination of false fire alarms. In this case fire detection by at least two smoke detectors turns on the external alarm (InstantTurnOnAlarm2). Fire detection by only one smoke detector raises the internal alarm (TurnOnAlarm1).

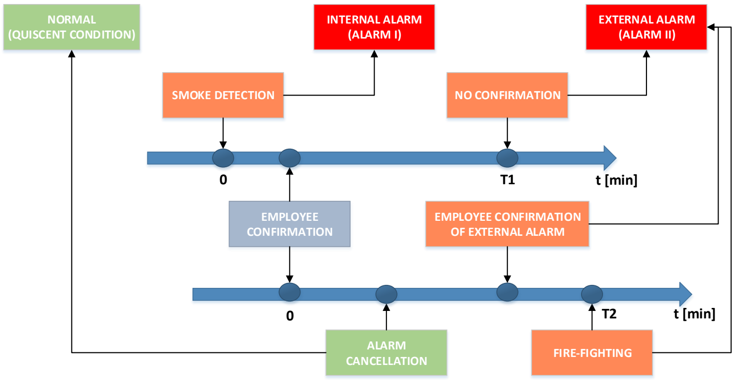
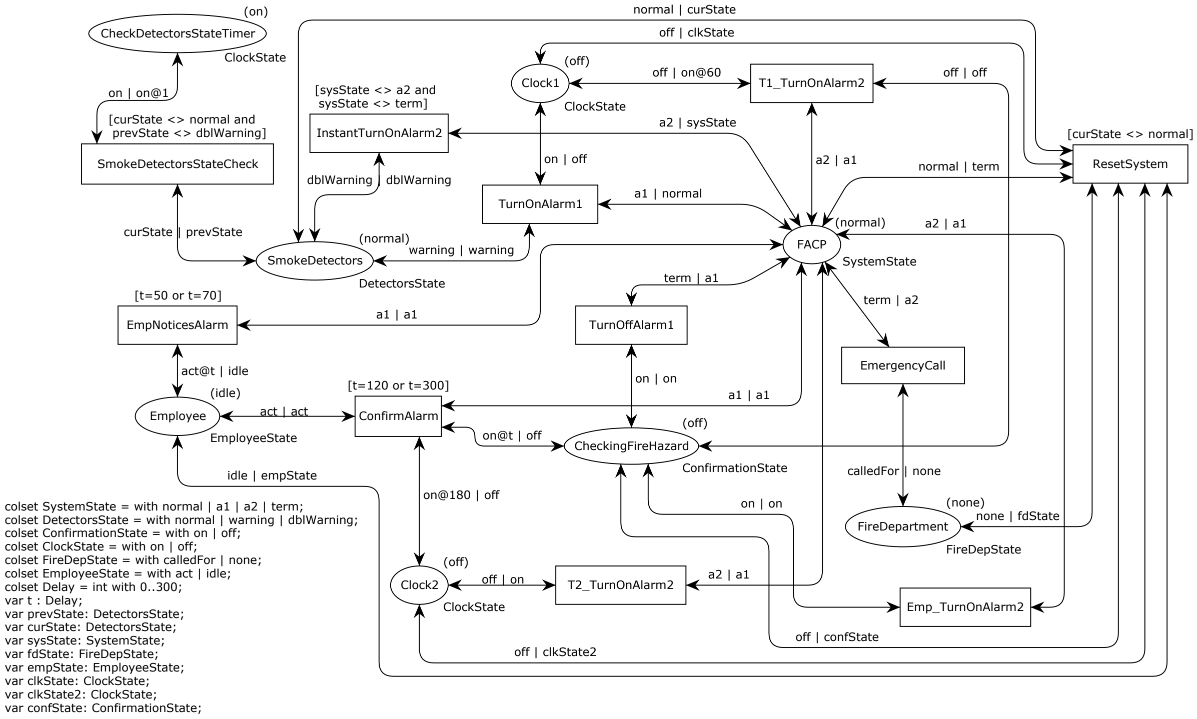


Figure 3. Fire alarm control panel [14].



```
colset SystemState = with normal | a1 | a2 | term;
colset DetectorsState = with normal | warning | dblWarning;
colset ConfirmationState = with on | off;
colset ClockState = with on | off;
colset FireDepState = with calledFor | none;
colset EmployeeState = with act | idle;
colset Delay = int with 0..300;
var t : Delay;
var prevState: DetectorsState;
var curState: DetectorsState;
var sysState: SystemState;
var fdState: FireDepState;
var empState: EmployeeState;
var clkState: ClockState;
var clkState2: ClockState;
var confState: ConfirmationState;
```

Figure 4. RTCP-net model of fire alarm control panel.

A fire alarm system is an example of a system which errors always cause major losses. False alarms generate high costs, but on the other hand any delays can lead to loss of human lives and health. Therefore, a comprehensive verification of such systems is of utmost importance. The coverability graph of the considered model has 3077 states and 3986 edges. Manual verification in this case is practically impossible. In the paper, the coverability graph of modelled system is automatically translated into Aldebaran format and verified using CADP Evaluator tool and μ-calculus formulae. Four examples of such formulae checking properties of the modelled system are presented below.

Listing 1: Examples of μ-calculus formulae for the model in Figure 4.

```
[true*."ConfirmAlarm".(not "ResetSystem")*."T1_TurnOnAlarm2".true*."ResetSystem"] false
[true*."ConfirmAlarm".(not "ResetSystem")*."TurnOffAlarm1".(not "ResetSystem")*."EmergencyCall"]
  ↪ false
[true*."InstantTurnOnAlarm2".(not "EmergencyCall")*."ResetSystem"] false
[true*."ConfirmAlarm".(not "ResetSystem" and not "T2_TurnOnAlarm2" and not "Emp_TurnOnAlarm2")*."
  ↪ EmergencyCall"] false
```

The first formula verifies whether acknowledgement of the internal alarm turns off the first timer, that is `Clock1`. Verification in CADP Evaluator proves, that even after expiration of `T1` time limit, the premature external alarm is not raised. The second formula checks whether operator's actions of confirmation and then internal alarm disarming are definitely preventing false external alarm from being raised. The third one denotes that fire detection by at least two smoke detectors always turns on the external alarm. These last two formulae are also proven true. The last formula denotes that after employee confirmation of the internal alarm, the external alarm cannot be raised unless the operator turns it on manually or timer `T2` expires. This formula, however, is not satisfied. Second smoke detector can possibly activate at any given moment, automatically turning on the external alarm, what was verified by the previous formula. It is also worth mentioning that a graph explaining the truth value of given formula can be generated using `-diag` option of the Evaluator tool.

SUMMARY

Action-based approach to verification of RTCP-nets has been presented in the paper. It employs conceived and implemented algorithm of translation of an RTCP-net coverability graph into the Aldebaran format. Employing PetriNet2ModelChecker tool, RTCP-nets modelled with CPN Tools can be automatically loaded and then, after the conversion, automatically verified with CADP Evaluator. To demonstrate the usability of this approach, an actual fire alarm control panel system has been modelled with the RTCP-net formalism. Some illustrative properties of the system have been specified using μ-calculus formulae and the results of the verification have been presented. Combined with the state-based verification introduced in [12] this approach allows comprehensive verification of any system modelled with the RTCP-net formalism.

REFERENCES

1. C. Baier, and J.-P. Katoen, *Principles of Model Checking*, The MIT Press, London, UK, 2008.
2. I. Grobelna, M. Wiśniewska, R. Wiśniewski, M. Grobelny, P. Mróz, "Decomposition, validation and documentation of control process specification in form of a Petri net", in *7th International Conference on Human System Interactions - HSI 2014*, Lisbon, Portugal, 2014, pp. 232–237.
3. R. Wiśniewski., A. Karatkevich, M. Adamski, D. Kur, "Application of comparability graphs in decomposition of Petri nets", in *7th International Conference on Human System Interactions - HSI 2014*, Lisbon, Portugal, 2014, pp. 216–220.
4. M. Szpyrka, "Analysis of RTCP-nets with Reachability Graphs", *Fundamenta Informaticae* **74**, 375–390 (2006).
5. M. Szpyrka, "Analysis of VME-Bus communication protocol – RTCP-net approach", *Real-Time Systems* **35**, 91–108 (2007).
6. K. Jensen, and L. Kristensen, *Coloured Petri nets. Modelling and Validation of Concurrent Systems*, Springer, 2009.
7. B. Jasiul, M. Szpyrka, and J. Śliwa, "Detection and Modeling of Cyber Attacks with Petri Nets", *Entropy* **16**, 6602–6623 (2014).
8. H. Garavel, F. Lang, R. Mateescu, and W. Serwe, "CADP 2006: A Toolbox for the Construction and Analysis of Distributed Processes", in *Computer Aided Verification*, Springer-Verlag, 2007, vol. 4590 of *LNCS*, pp. 158–163.
9. E. A. Emerson, "Model checking and the Mu-calculus", in *Descriptive Complexity and Finite Models*, edited by N. Immerman, and P. G. Kolaitis, American Mathematical Society, 1997, vol. 31 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 185–214.
10. D. Kozen, "Results on the propositional μ-calculus", *Theoretical Computer Science* **27**, 333–354 (1983).
11. R. Mateescu, and M. Sighireanu, Efficient on-the-fly model-checking for regular alternation-free μ-calculus, Tech. Rep. 3899, INRIA (2000).
12. A. Biernacka, J. Biernacki, and M. Szpyrka, "State-based verification of RTCP-nets with NuXMV", in *Proceedings of the Design and Analysis of Control Systems Conference (DACS 2015)*, Athens, Greece, 2015.
13. M. Szpyrka, A. Biernacka, and J. Biernacki, "Methods of translation of Petri nets to NuSMV language", in *Proceedings of the Concurrency Specification and Programming Workshop (CSP 2014)*, Chemnitz, Germany, 2014, vol. 1269 of *CEUR Workshop Proceedings*, pp. 245–256.
14. J. Ciszewski, K. Kunecki, W. Markowski, J. Sawicki, and M. Sobecki, *SITP Guideline WP-02:2010. Fire alarm systems. The design* (2010).