Model-Checking Based on Fluid Petri Nets for the Temperature Control System of the ICARO Co-generative Plant

Marco Gribaudo[1], A. Horváth[2], A. Bobbio[3], Enrico Tronci[4], Ester Ciancamerla[5], and Michele Minichino[5]

[1] Dip. di Informatica, Università di Torino, 10149 Torino, Italy
marcog@di.unito.it
[2] Dept. of Telecommunications, Univ. of Technology and Economics
Budapest, Hungary
horvath@webspn.hit.bme.hu
[3] Dip. di Informatica, Università del Piemonte Orientale, 15100 Alessandria, Italy
bobbio@unipmn.it
[4] Dip. di Informatica, Università di Roma "La Sapienza", 00198 Roma, Italy
tronci@dsi.uniroma1.it
[5] ENEA, CR Casaccia, 00060 Roma, Italy
{ciancamerla,minichino}@casaccia.enea.it

Abstract. The modeling and analysis of hybrid systems is a recent and challenging research area which is actually dominated by two main lines: a functional analysis based on the description of the system in terms of discrete state (hybrid) automata (whose goal is to ascertain for conformity and reachability properties), and a stochastic analysis (whose aim is to provide performance and dependability measures). This paper investigates a unifying view between formal methods and stochastic methods by proposing an analysis methodology of hybrid systems based on Fluid Petri Nets (FPN). It is shown that the same FPN model can be fed to a functional analyser for model checking as well as to a stochastic analyser for performance evaluation. We illustrate our approach and show its usefulness by applying it to a "real world" hybrid system: the temperature control system of a co-generative plant.

1 Introduction

This paper investigates an approach to model checking starting from a fluid Petri net (FPN) model, for formally verifying the functional and safety properties of hybrid systems. This paper shows that FPN [1, 11, 9] can constitute a suitable formalism for modeling hybrid systems, like the system under study, where a discrete state controller operates according to the variation of suitable continuous quantities (temperature, heat consumption). The parameters of the models are usually affected by uncertainty. A common and simple way to account for parameter uncertainty is to assign to them a range of variation (between a minimum and a maximum value), without any specification on the actual value

S. Anderson et al. (Eds.): SAFECOMP 2002, LNCS 2434, pp. 273–283, 2002.

assumed by the parameter in a specific realization (non-determinism). Hybrid automata [3] and discretized model checking tools [6] operate along this line. If a weight can be assigned to the parameter uncertainty through a probability distribution, we resolve the non-determinism by defining a stochastic model: the FPN formalism [11, 8] has been proposed to include stochastic specifications. However, the paper intends to show that a FPN model for an hybrid system can be utilized as an input model both for functional analysis as well as for stochastic analysis. In particular, the paper shows that the FPN model can be translated in terms of a hybrid automaton [2, 15] or a discrete model checker [5].

FPN's are an extension of Petri nets able to model systems with the coexistence of discrete and continuous variables [1, 11, 9]. The main characteristics of FPN is that the primitives (places, transitions and arcs) are partitioned in two groups: discrete primitives that handle discrete tokens (as in standard Petri nets) and continuous (or fluid) primitives that handle continuous quantities (referred to as fluid). Hence, in the single formalism, both discrete and continuous variables can be accommodated and their mutual interaction represented.

Even if Petri nets and model checking rely on very different conceptual and methodological bases (one coming from the world of performance analysis and the other form the world of formal methods), nevertheless the paper attempts to gain cross fertilizations from the two areas. The main goal of the research work presented in this paper is to investigate on the possibility of defining a methodology which allows to refer to a common FPN model to be used both for formal specification and verification with model checking tools and for performance analysis.

We describe our approach and show its usefulness by using a meaningful "real world" application. Namely, we assume as a case study the control system of the temperature of the primary and secondary circuit of the heat exchange section of the ICARO co-generative plant [4] in operation at centre of ENEA CR Casaccia. The plant, under study, is composed by two sections: the gas turbine section for producing electrical power and the heat exchange section for extracting heat from the turbine exhaust gases.

The paper is organized as follow. Section 2 describes our case study. Section 3 introduces the main elements of the FPN formalism, provides the FPN model of the case study, and its conversion into an hybrid automaton. Section 4 shows how the same FPN model can be translated into a discrete models checker (NuSMV [14]) and provides some of our experimental results. Section 5 gives the conclusions.

2 Temperature Control System

The ICARO co-generative plant is composed by two sections: the electrical power generation and the heat extraction from the turbine exhaust gases. The exhaust gases can be conveyed to a re-heating chamber to heat the water of a primary circuit and then, through a heat exchanger, to heat the water of a secondary circuit that, actually, is the heating circuit of the ENEA Research Center. If the thermal energy required by the end user is higher than the thermal energy of the exhaust gases, fresh methane gas can be fired in the re-heating chamber where the combustion occurs. The flow of the fresh methane gas is regulated by the control system through the position of a valve.

The block diagram of the temperature control of the primary and secondary circuits is depicted in Figure 1. The control of the thermal energy used to heat the primary circuit is performed by regulating both the flow rate of the exhaust gases through the diverter D and the flow rate of the fresh methane gas through the valve V. T_1 is the temperature of the primary circuit, T_2 is the temperature of the secondary circuit, and u is the thermal request by the end user.

The controller has two distinct regimes (two discrete states) represented by the position 1 or 2 of the switch W in Figure 1. Position 1 is the normal operational condition, position 2 is the safety condition. In position 1, the control is based on a proportional-integrative measure (performed by block PI_1) of the error of temperature T_2 with respect to a (constant) set point temperature T_s. Conversely, in position 2, the control is based on a proportional-integrative measure (performed by block PI_2) of the error of temperature T_1 with respect to a (constant) set point temperature T_s. Normally, the switch W is in position 1 and the control is performed on T_2 to maintain constant the temperature to the end user. Switching from position 1 to position 2 occurs for safety reasons, when the value of T_2 is higher than a critical value defined as the set point T_s augmented by an hysteresis value T_h and the control is locked to the temperature of the primary circuit T_1, until T_1 becomes lower than the set point T_s.

The exit of the proportional-integrative block (either PI_1 or PI_2, depending on the position of the switch W) is the variable y which represents the request of thermal energy. When y is lower than a split point value Ys the control just acts on the diverter D (flow of the exhaust gases), when the diverter is completely

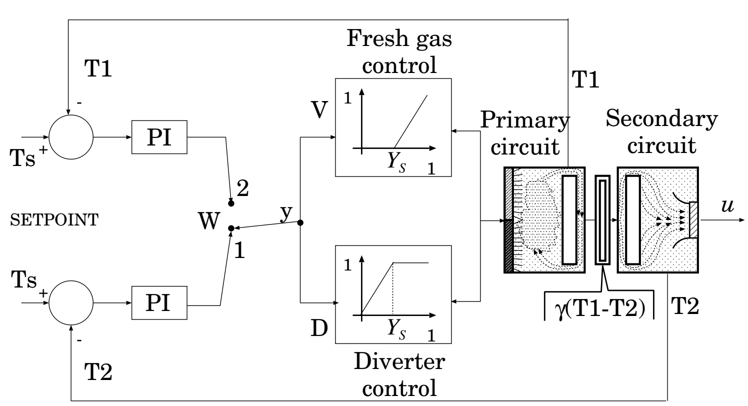

Fig. 1. Temperature control of the primary and secondary circuits of the ICARO plant

open, and the request for thermal energy y is greater than Ys, the control also acts on the flow rate of the fresh methane gas by opening the valve V.

The heating request is computed by the function $f(y)$ represented in Figure 2. Since the temperature T_2 is checked out when W is in position 1, and the temperature T_1 is checked out in state 2, the function $f(y)$ depends on y_2 when $\mathsf{W} = 1$ and on y_1 when $\mathsf{W} = 2$. The function $f(y)$ is defined as the sum of two non-deterministic components, namely: $g_1(y)$ which represents the state of the valve V, and $g_2(y)$ which represents the state of the diverter D. The non-determinism is introduced by the parameters $\alpha_{min}, \alpha_{max}$ that give the minimal and maximal heat induced by the fresh methane gas, and β_{min}, β_{max} that define the minimal and maximal heat induced by the exhaust gases.

Finally, the heat exchange between the primary and the secondary circuit is approximated by the linear function $\gamma(T_1 - T_2)$, proportional (through a constant γ) to the temperature difference.

3 Fluid Petri Nets

Fluid Petri Nets (FPN) are an extension of standard Petri Nets [13], where, beyond the normal places that contain a discrete number of tokens, new places are added that contain a continuous quantity (fluid). Hence, this extension is suitable to be considered for modeling and analyzing hybrid systems. Two main formalisms have been developed in the area of FPN: the Continuous or Hybrid Petri net (HPN) formalism [1], and the Fluid Stochastic Petri net (FSPN) formalism [11, 9]. A complete presentation of FPN is beyond the scope of the present paper and an extensive discussion of FPN in performance analysis can be found in [8].

Discrete places are drawn according to the standard notation and contain a discrete amount of tokens that are moved along discrete arcs. Fluid places are drawn by two concentric circles and contain a real variable (the fluid level). The fluid flows along fluid arcs (drawn by a double line to suggest a pipe) according to an instantaneous flow rate. The discrete part of the FPN regulates the flow of the fluid through the continuous part, and the enabling conditions of a transition depend only on the discrete part.

$$\frac{g_1(y)}{\alpha_{max}} + \frac{g_2(y)}{\beta_{max}} \le f(y) \le \frac{g_1(y)}{\alpha_{min}} + \frac{g_2(y)}{\beta_{min}}$$

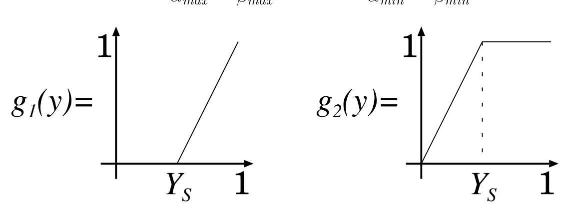

Fig. 2. The heating request function $f(x)$

3.1 A FPN Description of the System

The FPN modeling the case study of Figure 1 is represented in Figure 3. The FPN contains two discrete places: $P1$ which is marked when the switch W is in state 1, and $P2$ which is marked when the switch W is in state 2. Fluid place *Primary* (whose marking is denoted by T_1, and has a lower bound at T_s) represents the temperature of the primary circuit, and fluid place *Secondary* (whose marking is denoted by T_2 and has an upper bound at T_s+T_h) represents the temperature of the secondary. The fluid arcs labeled with $\gamma(T_1-T_2)$ represent the heat exchange between the primary and the secondary circuit. The system jumps from state 1 to state 2 due to the firing of immediate transition $Sw12$. This transition has associated a guard $T_2 > T_s + T_h$ that makes the transition fire (inducing a change of state) as soon as the temperature T_2 exceeds the *setpoint* T_s augmented by an histeresys value T_h. The change from state 2 to state 1 is modeled by the immediate transition $Sw21$, whose firing is controlled by the guard $T_1 < T_s$ that makes the transition fire when the temperature T_1 goes below the *setpoint* T_s. In order to simplify the figure, we have connected the fluid arcs directly to the immediate transitions. The meaning of this unusual feature is that fluid flows across the arcs as long as the immediate transitions are enabled regardless of the value of the guards.

The fluid arc in output from place *secondary*, represents the end user demand. The label on this arc is $[u_1, u_2]$, indicating the possible range of variation of the user demand. Fluid place $CTR2$, whose marking is denoted by y_1, models the exit of the proportional-integrator PI_1. This is achieved by connecting to place $CTR1$ an input fluid arc, characterized by a variable flow rate equal to T_2, and by an output fluid arc with a constant fluid rate equal to the *setpoint* T_s. In a similar way, the exit of the proportional-integrator PI_2 is modeled by fluid place $CTR2$ (whose marking is denoted by y_2). The fluid arcs that connect transition $Sw12$ and $Sw21$ to fluid place *primary* represent the heating up of the primary circuit.

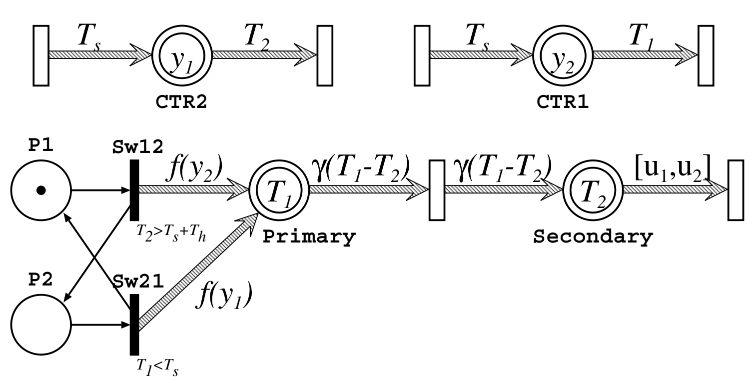

Fig. 3. FPN model of the temperature Controller

3.2 From FPN to Hybrid Automata

An hybrid automaton [3]is a finite state machine whose nodes (called control modes) contain real valued variables with a definition of their first derivatives and possible bounds on their values. The edges represent discrete events and are labeled with guarded assignments on the real variables. Given a hybrid automaton and a legal formula on its variables, the *model checking problem* asks to compute a region that satisfies the predicate, or to find at least one counterexamples that contradicts the predicate. In order to use a FPN model in a model checking environment, the FPN formalism sould be converted into a hybrid automaton. A general conversion algorithm could be envisaged using the technique proposed in [15], and some of the ideas presented in [2]. The application of the general algorithm to the case study FPN of Figure 3 provides the hybrid automaton [3] of Figure 4.

The hybrid automaton has the following set of real variables T_1, T_2, y_1 and y_2 (corresponding to the fluid variables of the FPN) and two control modes $P1$ and $P2$ (corresponding to the two discrete markings of the FPN). Each continuous variable has a derivative equal to the flow rate of the corresponding fluid place in that state. Transitions from control mode $P1$ to $P2$ and from $P2$ to $P1$ are labeled with the guards of the immediate transitions that cause the state change. State $P1$ has also associated the bound (invariant condition) $T_2 \leq T_s + T_h$ and $P2$ the bound $T_1 \geq T_s$ to reflect the same bounds posed on those fluid places. The model of Figure 4 could be analyzed by means of appropriate tools for hybrid automata [10].

4 Analysis of the FPN Model via NuSMV

Discrete model checking is based on a finite state machine model in which the variables and their derivatives are discretized and in which the time increases with a predefined time step. The parameters and their derivatives can be assigned uncertainty ranges (e.g. a min and a max value) with non-deterministic logic. The predicates to be checked are specified using a Computational Tree Logic (CTL) or a Real Time CTL (RTCTL) [6, 7].

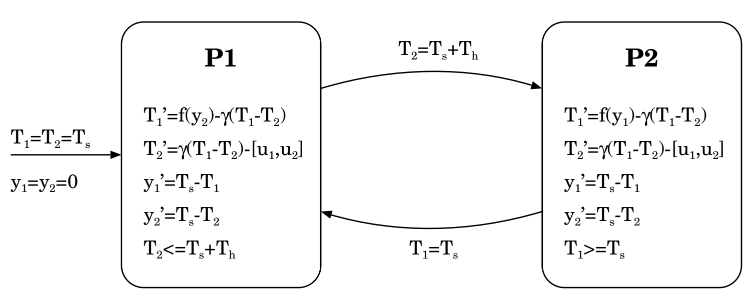

Fig. 4. Hybrid Automata obtained from the FPN of Figure 3

In order to show the generality of our approach and to give an insight on the class of models that can be automatically derived from the FPN description, we sketch, in brief, how the FPN can be converted into a discrete model to be checked using discrete model checking techniques and we present some typical analyses and results that can be obtained from the converted model. For the purpose of the present paper, we have chosen the language NuSMV, for which an analysis tool is available [14].

4.1 Converting a FPN into a NuSMV Model

In the present section, we describe the main steps that are required to convert a FPN model into the NuSMV language, and we provide an excerpt of the NuSMV specifications for the case study at hand, in the Appendix. A more detailed description of the conversion algorithm and the complete NuSMV specifications are given in [12]. The conversion algorithm requires the following steps:

1. Definition of the variables. The discrete part of the FPN model is the marking and it is directly translated into a discrete variable in NuSMV. All the continuous variables (fluid levels of the FPN) and their rates of variation must be, instead, suitably discretized. These facts are described in NuSMV under the keyword `VAR`.
2. The second step requires the FPN constants that are used in the fluid rate functions or in the enabling conditions are defined. Moreover, the ranges of variation (min and max values) for the continuous variables and for their rates must be set. All these quantities (constants and bounds) must be suitably discretized and rescaled according to the discretization intervals chosen in step 1), above. The constants and the bounds are listed under the keyword `DEFINE`.
3. In order to analyze the behavior of the control system versus time, a time step (in arbitrary units) is assumed and the dynamic evolution of the system at the integer multiples of the time step must be described. The evolution of the model is stated under the keyword `TRANS` and must be described marking by marking.
4. Finally, the fourth step consists in defining an initial state from which the dynamic evolution of the model starts. The initial state of the model is described under the keyword `INIT`.

We now particularize the above general points to the present case study (refer to the Appendix and to [12] for the obtained specification in the NuSMV language). The discrete part of the FPN model is reflected in the variable *marking*, whose value is either 1 or 2. Furthermore, all the continuous variables (fluid levels of the FPN) and their range of variation must be discretized. Let x be a fluid variable in FPN whose fluid place is lower bounded by B_l and upper bounded by B_u. We define a discretization step δ such that the continuous range of variation of x is discretized in n steps with $n = \lceil (B_u - B_l)/\delta \rceil$ ($\lceil \cdot \rceil$ denotes the closest larger integer of its argument). With this assumption, the possible discretized values of the level x are defined in NuSMV as `x: 0..n` where $x = i$ means that the corresponding value is $B_l + i\delta$.

In the FPN of Figure 3, four fluid variables are defined: y_1, y_2 and T_1, T_2. In the NuSMV description, the variables representing the fluid levels y_1, y_2, T_1 and T_2 are denoted by `y1`, `y2`, `T1`, and `T2`. The fluid levels y_1 and y_2, of fluid places $CTR1$ and $CTR2$, respectively, are normalized in the range $[0, 1]$, and discretized with a step interval $1/30$. The normalization constant for y_1 and y_2 is denoted by dy and represents how fast the system reacts to the temperature difference with respect to the setpoint. The fluid levels T_1 and T_2 of fluid places $Primary$ and $Secondary$, respectively, are bounded between $T_\ell = 138$ and $T_u = 145$ and the discretization step chosen for these variables is 0.1.

The list of constants and bounds defined in step 2 above, includes:

– `y1max`, `y2max`,`T1max`, `T2max` rescaled bounds on the continuous variables;
– `alphamin, alphamax,`and `betamin, betamax`: non-deterministic range of the heat induced by the methane gas and by the exhaust gas, respectively;
– `sp`, `hys`, `ys`: setpoint temperature T_s, hysteresis value T_h and split point Y_s;
– `dy`: system reaction speed to the output of the proportional integrators PI;
– `gamma` rate of heat exchange between the primary and secondary circuit;
– `u1, u2`: non-deterministic range in the heat consumption of the end user.

The second part under the keyword `DEFINE` defines the rates at which the continuous variables change in each discrete state, and the bound on the rates. For `marking=1`, these are the following (similar definition hold for `marking=2`):

– $m1\_y1$ gives the (deterministic) fluid rate of place $CTR1$ in state 1;
– $m1\_y2$ gives the (deterministic) fluid rate of place $CTR2$ in state 1;
– $m1\_T1\_min$ and $m1\_T1\_max$ give the minimal and maximal flow rate of fluid place $Primary$ in state 1;
– $m1\_T2\_min$ and $m1\_T2\_max$ give the minimal and maximal flow rate of fluid place $Secondary$ in state 1;

Since in the present model we have two markings (states), the evolution description is restricted to four expressions:

– possible changes of the variables inside `marking=1` (`marking=2`) ;
– jump from `marking=1` to `marking=2` (from `marking=2` to `marking=1`).

Finally, the initial state of the model is described under the keyword `INIT`.

4.2 NuSMV Results

NuSMV is a model checking tool and contains also a simulation engine to explore the dynamics of the system. To increase the readability of the results, we report the variables in their true units (and not in the rescaled units used by NuSMV). Figure 5 and 6 depict the evolution of the temperatures (`T1` and `T2`) and of `y1` and `y2`, respectively, for the same simulation trace, starting from the initial state `[y1=0, y2=0, T1=138, T2=138]`.

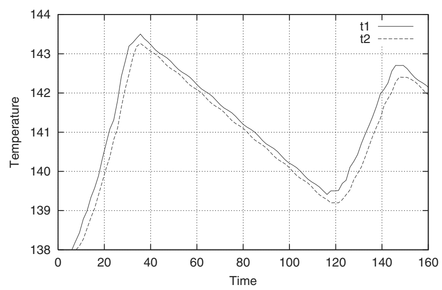

Fig. 5. Change of temperature given by a simulation trace

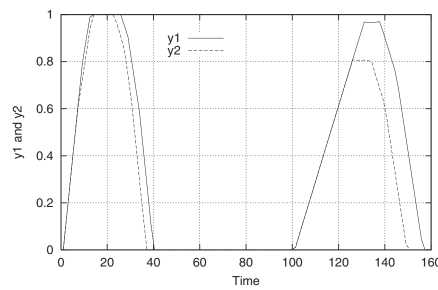

Fig. 6. Change of y_1 and y_2 given by a simulation trace

The design specification for the temperatures of the system (given as invariant condition or bounds) are: ($139 \leq$ `T1` ≤ 144 and $139 \leq$ `T1` ≤ 141). If the invariant does not hold, i.e. the temperatures exceed the bounds, NuSMV produces a counterexample as in Table 1. The table shows a case with `[gamma=2, dy=10, Ts=140]`. Both temperatures start initially from 141 and decrease because of the heat consumption of the user. As `T2` (`T1`) reaches `Ts`, `y1` (`y2`) starts to increase. However, the reaction is not fast enough to avoid the undesirable condition and the secondary temperature `T2` crosses the lower bound. Modifying the design parameters may avoid to incur in this situation (f.i. setting `[gamma=2, dy=1/10, Ts=139.8]`, i.e. speeding up the reaction of the system, and reducing the setpoint temperature).

Using RTCTL (Real-Time Computational Tree Logic [7]) expression, one can check the trajectory on which the system proceeds. For example, starting from the lowest possible temperatures (`T1=T2=138`) the formula

```
AF (AG (T1>=139 & T1<=144 & T2>=139 & T2<=141))
```

is true if the system gets back to stable state for sure and remains there forever. Setting `[gamma=2, dy=1/10, sp=18]`, the formula evaluates to true. The same formula, with the same settings evaluates to true as well, if the system starts from the upper bound of the temperatures.

Knowing the timing behavior of the system, one can use NuSMV to compute the minimal or maximal time needed to get a given set of states from an initial condition. For example, the command

```
COMPUTE MIN[y1=0 & y2=0 & T1=145 & T2=145, AG (T1>=139 & T1<=144 &
```

Table 1. Counterexample

Step	1	2	3	4	5	6	7	8	9	10	11	12	13	14
State	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T1	141	141	140.9	140.7	140.6	140.4	140.3	140.1	140	139.8	139.7	139.5	139.4	139.2
T2	141	140.7	140.5	140.4	140.2	140.1	139.9	139.8	139.6	139.5	139.3	139.2	139	138.9
y1	0	0	0	0	0	0	0	1/30	2/30	3/30	4/30	5/30	6/30	8/30
y2	0	0	0	0	0	0	0	0	0	1/30	2/30	3/30	4/30	5/30

```
T2>=139 & T2<=141)]

COMPUTE MAX[y1=0 & y2=0 & T1=145 & T2=145, AG (T1>=139 & T1<=144 &
T2>=139 & T2<=141)]
```

gives the length of the minimal and maximal paths that lead from the initial condition `[y1=0, y2=0, T1=145, T2=145]` of high temperatures (out of the required range) to temperatures inside the required range in such a way that the system does not leave this range in the future. The above command with parameters `[gamma=2, dy=1/10, sp=18]` results in $min-path = 21$ and $Max-path = 64$.

5 Scalability and Complexity

The scalability and the complexity of the method mainly depends on the hybrid automata solution component. The process of translating a FPN to Hybrid System is exponential in the dimension of the FPN: it requires the creation of its reachability graph which is done through a depth-first visit of its state space. This step is clearly exponential in the dimension of the model (see for example [9]). After the model has been translated, the complexity of the analysis depends on the complexity of the algorithms used by the NuSMV package.

The scalability of the technique is thus limited by two different aspects: the exponential complexity of the translation process, and the solution complexity of the Hybrid Automata analysis technique. At the present time, these constraints limit the applicability of the proposed technique only to very small (in term of FPN description elements) models.

6 Conclusion

Using a real world hybrid system as a case study we presented an approach to integrate FPNs and model checking via hybrid automata and NuSMV.

Such integration turns out to be conceptually useful and effective in practice. In fact it allowed us to comfortably model and verify the temperature control system in the co-generative plant ICARO at ENEA (CR).

Acknowledgment

A. Bobbio and M. Gribaudo were partially supported by the Italian Ministry of Education under Grant ISIDE.

References

[1] H. Alla and R. David. Continuous and hybrid Petri nets. *Journal of Systems Circuits and Computers*, 8(1):159–188, Feb 1998. 273, 274, 276

[2] M. Allam. Sur l'analyse quantitative des réseaux de Petri hybrides: une approche baseée sur lea automates hybrides. Technical report, Phd Thesis, Institut National Polytechnique de Grenoble (in French), 1998. 274, 278

[3] R. Alur, T. A. Henzinger, and P. H. Ho. Automatic symbolic verification of embedded systems. *IEEE Transaction Software Engineering*, 22:181–201, 1996. 274, 278

[4] A. Bobbio, S. Bologna, E. Ciancamerla, P. Incalcaterra, C. Kropp, M. Minichino, and E. Tronci. Advanced techniques for safety analysis applied to the gas turbine control system of ICARO co-generative plant. In *X Convegno Tecnologie e Sistemi Energetici Complessi*, pages 339–350, 2001. 274

[5] A. Bobbio and A. Horváth. Petri nets with discrete phase timing: A bridge between stochastic and functional analysis. In *Second International Workshop on Models for Time-Critical Systems (MTCS 2001)*, pages 22–38, 2001. 274

[6] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986. 274, 278

[7] E. A. Emerson, A. K. Mok, A. P. Sistla, and J. Srinivasan. Quantitative Temporal Reasoning. *Journal of Real Time Systems*, 4:331–352, 1992. 278, 281

[8] M. Gribaudo. Hybrid formalism for performance evaluation: Theory and applications. Technical report, Phd Thesis, Dipartimento di Informatica, Università di Torino, 2001. 274, 276

[9] M. Gribaudo, M. Sereno, A. Horváth, and A. Bobbio. Fluid stochastic Petri nets augmented with flush-out arcs: Modelling and analysis. *Discrete Event Dynamic Systems*, 11 (1/2):97–117, January 2001. 273, 274, 276, 282

[10] T. A. Henzinger, P. H. Ho, and H. Wong-Toi. A user guide to HyTech. In *Proceedings 1st Workshop Tools and Algorithms for the Construction and Analysis of Systems - TACAS*, pages 41–71. Springer Verlag, LNCS Vol 1019 - http:// www.eecs.berkeley.edu/ tah/HyTech, 1995. 278

[11] G. Horton, V. Kulkarni, D. Nicol, and K. Trivedi. Fluid stochastic Petri nets: Theory, application and solution techniques. *European Journal of Operational Research*, 105(1):184–201, 1998. 273, 274, 276

[12] A. Horváth, M. Gribaudo, and A. Bobbio. From FPN to NuSMV: The temperature control system of the ICARO cogenerative plant. Technical report, Università del Piemonte Orientale, Feb 2002, `http://www.di.unipmn.it`. 279

[13] T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77:541–580, 1989. 276

[14] *NuSMV*. `http://nusmv.irst.itc.it/index.html`. 274, 279

[15] B. Tuffin, D.S. Chen, and K. Trivedi. Comparison of hybrid systems and fluid stochastic Petri nets. *Discrete Event Dynamic Systems*, 11 (1/2):77–95, January 2001. 274, 278