On Quantitative Assessment of Reliability and Energy Consumption Indicators in Railway Systems

Davide Basile, Felicita Di Giandomenico, and Stefania Gnesi

Abstract Stochastic model-based approaches are widely used for obtaining quantitative non functional indicators of the analysed systems, as for example reliability, performance and energy consumption. However, a critical issue with models is their validation, in order to justifiably put reliance on the analysis results they provide. In this paper, we address cross-validation on a case study from the railway domain, by modelling and evaluating it with different formalisms and tools. Stochastic Activity Networks models and Stochastic Hybrid Automata models of rail road switch heaters, developed for the purpose of evaluating energy consumption and reliability indicators, will be evaluated with Mobius and Uppaal SMC. We will compare the obtained results, to improve their trusthworthiness and to provide insights on the design and analysis of energy-saving cyber-physical systems.

Keywords: energy-saving, reliability, quality models, stochastic analysis

1 Introduction

Recently, studies dedicated to reduce the energy consumption in cyber-physical systems (CPS) [41] are gaining increasing attention [37, 36, 35, 25, 28, 12], aiming at saving in economic terms and reducing environmental impact. In CPS, digital control units interact with phenomena belonging to the system itself or the surrounding environment, whose nature is typically continuous (e.g., failure events, power energy flow). Examples of CPS can be found in disparate application domains, including the transportation sector. The continuous dynamic nature of CPS is difficult to be expressed through discrete approaches, with proper control on the required approximations. Extensions of finite state automata [3, 31, 38], (extensions of) Petri Nets [7, 48, 23], have been adopted as formalisms for modelling them, where the

Institute of Information Science and Technologies (ISTI), National Research Council (CNR), Pisa, e-mail: {davide.basile,felicita.digiandomenico,stefania.gnesi}@isti.cnr.it

evolution of the continuous variables can be described uniformly or by ordinary differential equations.

Dependability evaluations and formal verification are two separate research fields that in the last decade have been integrated to analyse critical CPS, where (i) measures as dependability, performance, reliability, energy consumption are formally assessed through rigorous approaches, and (ii) the correctness of the models is ascertained by proving that relevant properties hold in a suitable abstraction of the analysed system. These two aspects are complementary, and to cope with potential defects introduced in the modelling phase, validation of the developed models is paramount and highly recommended. Indeed, the introduced errors might compromise the accuracy of the results obtained through the analysis, which may lead to the delivery of flawed components, with both potential serious consequences for the components users and loss of time and money for industries, to recover from the late revealed deficiencies.

In this paper, we compare these two approaches for the analysis of CPS with reference to a case study from the railway domain. The aim of the investigation is to (i) emphasise their differences and provide considerations for modelling and evaluating CPS, and (ii) to verify that the results obtained through two separate formalisations of the case study are indeed in accordance.

Specifically, we consider rail road switch heaters, which are essential components for the correct functioning of railway stations, in absence of which possible disasters can occur (i.e. derailments, trains collision). In particularly cold regions, ice and snow can prevent the switches to work properly, hence heaters are used for guaranteeing their correct functioning. In particular, a central control unit is in charge of managing policies of energy consumption while satisfying reliability constraints, by communicating with the network of switches to manage the energy supply.

We will exploit two different methodologies to model and evaluate the system under analysis, chosen among popular ones in the research communities interested in verification and evaluation of complex critical systems. The first is based on the adoption of *Stochastic Activity Networks* (SAN) [48] to model rail road switch heaters for assessing the energy consumption and the probability of failure of these devices [11,10,9]. The quantitative properties (or measures of interest) are defined using Markov Reward Models [30] and evaluated in Möbius [18] through simulation.

The second adopted methodology is based on *Statistical Model Checking* (SMC) [42, 39]. The system is modelled as a network of *Stochastic Hybrid Automata* (SHA) [31,38, 22] to uniformly deal with both continuous, discrete and stochastic aspects. The quantitative properties are defined in the Metric Interval Temporal Logic (MITL) [22] and evaluated with Statistical Model Checking [14]. The Uppaal SMC toolkit [22] is used for evaluating these measures. Moreover, the absence of deadlocks has been proved through standard model checking techniques and through refinement[13]. Those models are here extended to deal with different classes of priorities in the network of switches.

The main contributions of this paper are:

- modelling frameworks of rail road switch heaters through both SAN and SHA formalisms, in order to compare different strategies of energy consumption, considering failure events and different weather profiles;
- identification of indicators representative of energy consumption and reliability of the analysed system and their quantification through the developed frameworks in a variety of scenarios. The obtained results allow to gain insights on suitable tradeoffs between energy consumption and reliability of the analysed system, to properly tune the parameters of the considered energy consumption strategies;
- comparing the results obtained through the two separate formalisations provides a further guarantee on the reliability of the obtained results.

Structure of the paper The paper is structured as follows. We will introduce dependability analysis and the two methodologies that we will adopt in Section 2. In Section 3 we will describe the system that we intend to model and analyse, with the goal of reducing the energy consumption and augmenting the reliability of the system. Generic guidelines for modelling energy-saving CPS are discussed in Section 4. The two different formalisations of the system are in Section 5, while the evaluation and validation of these models are in Section 6. Finally, related work and conclusions are respectively in Section 8 and Section 9.

2 Dependability Analysis

Dependability analysis of complex, critical systems is the topic of many research studies, since assurance on proper operation is typically among the requirements of the applications such systems are employed in. Depending on the specific aspects of interest in the dependability field, different approaches have been developed to accomplish analyses. The most active communities in dependability analyses have been and are: (i) the one focusing on fault tolerance and dependability, and (ii) the one focusing on formal methods. Although moving from rather different perspectives, both are progressively converging towards solution of common problems and in the last decade they influenced each other [6].

In particular, due to the emerging complexity and dependencies shown by modern systems, like many critical CPS around our everyday life, there is a need for expressing more complex system behaviours and measures of interest. This motivated growing relevance of quantitative measures in the field of verification, while in the past qualitative aspects were mainly tackled. Quantitative assessment of non-functional properties, especially dependability and performance related ones, consists in probabilistically estimate the occurrence of faults and their impact on the ability of the system to operate correctly.

To provide the context necessary to better understand the motivation of our study, in this section we briefly overview two main approaches to quantitative dependability analyses developed by the traditional dependability and the formal methods communities. Namely, quantitative model-based analysis and Statistical Model Checking are introduced in the following, together with the formalisms and tools that we will adopt in this paper.

2.1 Model-based Dependability and Performance Analysis

Several approaches are available in the literature to perform assessment of dependable systems [5], mainly testing, fault injection and model-based evaluation. A wide range of modelling techniques has been developed for both dependability and security analysis, each focusing on particular levels of abstraction and/or system characteristics, as surveyed in [49].

For quantitative evaluation of dependability indicators, stochastic model-based analysis [51] has been proven to be particularly useful, versatile and cost-effective for manufacturers [16,24]. The system under analysis is often described through stochastic processes, whilst measures of interest are generally obtained through mathematical analysis (closed-form expressions), numerical evaluation (linear programming techniques) and discrete-event simulation (statistical methods). This analysis approach is useful for expressing the stochastic nature of physical phenomena involved in CPS [41].

To keep the model manageable, the system needs to be represented at a properly identified abstraction level. Indeed, depending on the properties to be analysed the emphasis on the system representation is focused on those aspects that are relevant for analysis purposes, while irrelevant aspects are neglected. Therefore, a wide variety of models are used in practice, to tailor the right abstraction level for the system under analysis, in accordance with the properties to be assessed, the desired degree of accuracy and available resources to manage models development and solution. Formalisms such as (extensions of) Petri Nets [7,48,23] and (Non) Markov based models [48,15] are used for modelling and evaluating CPS, where reward structures [47] are defined in order to evaluate measures of interest (e.g. reliability, performance, energy consumption) at the variation of relevant parameters, either analytically or through simulation. Stochastic Activity Networks [48] and Möbius [18] which we adopted in our study, are well-established formalism and tool, respectively, for modelling and evaluating these systems.

Stochastic Activity Network Stochastic Activity Networks (SAN) [48] is a formalism widely used for performance, dependability and performability evaluation of complex systems, given its high expressiveness and the powerful tools for modelling and evaluating them [18]. The SAN formalism is a variant of Stochastic Petri Nets [15], and

has similarities with Generalised Stochastic Petri Nets [7]. A SAN is composed of the following primitives: *places, activities, input gates* and *output gates*. Places and activities have the same interpretation as places and transitions of Petri Nets. Input gates control the enabling conditions of an activity and define the change of marking when an activity completes. Output gates define the change of marking upon completion of the activity. Activities are of two types: *instantaneous* and *timed*. Instantaneous activities complete once the enabling conditions are satisfied. Timed activities take an amount of time to complete following a temporal stochastic distribution function, which can be for example exponential or deterministic. When an activity completes, the following steps are executed: (1) one of the cases of the activity is chosen according to its markingdepending probability; (2) the function of each input gate of the activity is executed; (3) the function of each output gate linked to the case selected at first step is executed. An enabled activity is aborted, i.e. it cannot complete, when the SAN moves into a new marking in which the enabling conditions of the activity no longer hold. Cases are associated to activities, and are used to represent probabilistic uncertainty about the action taken upon completion of the activity.
The primitives of the SAN models are defined using C++ code.

Möbius Möbius [18] is a multi-formalism multi-solver tool that can be used for defining and solving SAN models. Möbius supports various formalisms and different analytical and simulative solvers, and can be used for studying the reliability, availability, and performability of systems. It follows a modular modelling approach, where atomic models are building blocks that can be composed with proper operators *Rep* and *Join* to generate a composed model.

Notably, atomic models specified in different formalisms can be composed in this way. This allows to specify different aspects of a system under evaluation in the most suitable formalism. Along with an atomic or composed model, the user specifies a reward model, which defines a reward structure on the overall model. Rewards are the vehicle to define the measures for our case study. On top of a reward model, the tool provides support to define experiment series, called Studies, in which the user defines the set of input parameters for which the composed model should be evaluated. Each combination of input parameters defines a so-called experiment. Before analysing the model experiments, a solution method has to be selected: Möbius offers a powerful (distributed) discrete-event simulator, and, for Markovian models, explicit state-space generators and numerical solution algorithms. It is possible to analyze transient and steady-state reward models. The solver solves each experiment with parameters instantiation specified in the Study. Results can be managed by means of a database.

2.2 Formal Methods and Statistical Model Checking

Recent developments in probabilistic analysis using formal methods have improved the accuracy and reliability of dependability analysis, which was traditionally performed through proof methods not fully automatised and computer simulations [2]. In the literature, several approaches for the verification and validation of stochastic models have been proposed, as for example testing, theorem proving, model checking. In particular, model checking [19] is a widely-used and powerful approach for the verification of finite state systems, where a property φ, usually specified in a temporal logic, is automatically checked against a model of a system M, by performing an exhaustive exploration of the state-space of M, i.e. $M \models \varphi$, obtaining a counter-example in case $M \models \varphi$ does not hold.

The recently introduced Statistical Model Checking [42,39] uses results from statistics on top of simulations of a system to decide whether a given property specified in a temporal logic is satisfied with some degree of confidence, and it represents a valid alternative to classical model checking and dependability evaluation, especially in the case of undecidability. PRISM [33] and Uppaal SMC [22] have been proposed as tools that implement the above techniques. An advantage is that quantitative properties are uniformly described through temporal logics, and hence have a well-defined semantics. Moreover, it is possible to assess qualitative as well as quantitative properties to evaluate and validate the proposed models.

Stochastic Hybrid Automata, analysed with Uppaal SMC is the specific model checking approach adopted in this paper to address the modelling and analysis of our case study.

Hybrid Automata Stochastic Hybrid Automata (SHA) [31,38,22] are a suitable formalism for describing cyber-physical systems, where both discrete and continuous dynamics and stochastic phenomena are involved. Timed automata [3] combine discrete systems with real-valued variables that evolve during the time a system spends in a state. These variables, called *clocks*, evolve with a uniform rate and they can be used for guarding transitions. Reachability and other key problems are decidable for timed automata, with algorithms supporting them implemented in tools such as Uppaal [40].

Hybrid automata [31,38] generalise timed automata by including arbitrary dynamics for the real-valued variables (i.e. clocks), expressed through ordinary differential equations (ODEs). Stochastic Hybrid Automata include also probabilistic transitions and are used in tools such as Uppaal SMC [22]. A number of case-studies demonstrate their applications [20] [22].

For simplifying the presentation, we slightly elaborate the formal definition of hybrid automata in [31,38]. We start by introducing some useful notation. In a hybrid automaton the states progress according to both continuous and discrete clocks, in the first case this behaviour is called *continuous flow*, while in the second *jump*. A flow function $R^{|X|} \to R^{|X|}$ characterises the flow (i.e. the dynamic) of the continuous variables in the set X through a system of ODEs $\dot{X} = F(X)$, where $\dot{X}$ is the first order derivatives of the variables in X, and as usual R is the set of real numbers. Moreover, let $v : X \to R$ be a valuation of the variables in X, $\pi \in \mathrm{pred}(X)$ be a predicate over X and $[[\pi]] \in R^{|X|}$ be the set of valuations of X that satisfies the predicate π. Predicates are used to (*i*) guard transitions, (*ii*) specify the jumps of a system (i.e. how variables evolve in a discrete-time step) and (*iii*) define the invariants for each state of the automaton. A hybrid automaton H is defined as a tuple $H = <Q, Q_0, \Sigma, X, T, I, F, V_0>$ where:

- Q is a finite set of *states* including a distinguished initial singleton set $Q_0 \subseteq Q$,
- Σ is a finite set of *actions*,
- X is a finite set of real-valued *variables*, called *clocks*,
- $\mathrm{T} \subseteq Q \times \mathrm{pred}(X) \times \Sigma \times \mathrm{pred}(X \cup X^0) \times Q$ is the *transition relation*,
- $I : Q \rightarrow \mathrm{pred}(X)$ that assigns an invariant function to each state,
- $F : Q \rightarrow (\mathrm{R}^{|X|} \rightarrow \mathrm{R}^{|X|})$ that assigns a flow function to each state $q \in Q$ as the set of ODEs $X = F(q)(X)$, and
- $V_0 \in \mathrm{pred}(X)$ is the set of initial valuations.

It is assumed that for each state $q \in Q$ the flow function $F(q)$ has unique solution. We now briefly describe the semantics of hybrid automata. A configuration of a hybrid automaton is a tuple (q,v) where $q \in Q$ is a state and $\mathrm{v} \in \mathrm{R}^{|X|}$ is a variable valuation.

The initial configuration of a hybrid automaton is (q_0,v_0), where $q_0 \in Q_0$, $\mathrm{v}_0 = [[\pi]]$ such that $\pi \in V_0$ and $\mathrm{v}_0 \in [[I(q_0)]]$ (the invariant constraints are satisfied). During the time t a system spends in a state q, the clocks in X are updated according to the flow function of q, and at each step the new valuation must respect the invariant constraints in q. A transition $\delta = (q,g,a, j,q_1)$ is enabled after t time when the guard $g \in \mathrm{pred}(X)$ is satisfied. When δ is executed, the automaton jumps to a new configuration (q_1,v_1) such that q_1 is the target state of δ, v_1 is the valuation of the jump constraints $j \in \mathrm{pred}(X \cup X^0)$, and $\mathrm{v}_1 \in [[I(q_1)]]$.

Composing Hybrid Automata For modelling complex hybrid systems it is convenient to adopt a modular approach where systems are described by interacting entities. This allows to separately verify different smaller components more efficiently than verifying a bigger monolithic model. Hybrid automata can be composed through a synchronous product operator, and they interact through actions and shared variables. Let $I = \{1,\ldots,n\}$ be a set of indexes, the product of hybrid automata is denoted as $\bigotimes_{H_i \in C} H_i$, where $C = \{H_i \mid i \in I\}$. The states of the product are composed by the product of the states of its components. Similarly, the alphabet and the variables are the union of those of its components. The invariants, flow function and initial valuations are defined homomorphically on their elements. Finally, the transitions are synchronous, i.e. all the components (satisfying the constraints on the corresponding transition) synchronise on an action a while the others stay idle (in the following we will also distinguish between input and output actions through broadcast channels).

Uppaal SMC Uppaal is a toolbox that has been adopted for verifying real-time systems, represented by (extended) timed automata, that interact through broadcast channels and shared variables. Uppaal SMC is an extension of Uppaal that allows to express both stochastic and non-linear dynamic features, by adopting a stochastic extension of hybrid automata. The stochastic interpretation replaces the non-deterministic choices for multiple enabled transitions and time delays with, respectively, probabilistic choices and probability distributions (uniform for bounded time and exponential for unbounded time). By composing different automata through the product of SHA, arbitrary complex behaviours can be obtained, where it is possible to statically or dynamically generate new instances of automata, that are uniquely identified.

Uppaal SMC uses Statistical Model Checking to evaluate probabilistic properties of interest. SMC uses results from statistic area to decide, based on a given number of monitored simulations, whether the system under analysis satisfies the property of interest within a given degree of confidence. An advantage of SMC is that it avoids the exploration of the whole state-space of a model, which is a main drawback of standard model checking techniques.

Temporal Logic formulae. In addition to standard model checking techniques of properties as reachability, deadlock-freedom, in Uppaal SMC it is possible to evaluate the probability that a random run of a network M satisfies a property φ in a given amount of time t. Properties are defined using the Metric Interval Temporal Logic (MITL) [22]. A MITL formula φ is inductively defined by the following grammar:

$$\varphi ::= \mathrm{ap} \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \cup^{x}_{\leq t} \varphi_2$$

In the definition above, ap are atomic predicates over states of an automaton, and the logical operators are standard, except for $\varphi_1 \cup^{x}_{\leq t} \varphi_2$ that checks whether a formula φ_1 is satisfied in a run *until* a formula φ_2 is satisfied, and this must happen before the clock x exceeds the value t. As usual, it is possible to derive the operators *exists* and *forall* as $\Diamond_{x\leq t}\varphi = \text{true} \bigcup^{\mathrm{x}}_{\leq t} \varphi$ and $\Box_{x\leq t}\varphi = \neg\Diamond_{x\leq t}\neg\varphi$, where both quantifiers are bounded by the time t for the clock x.

Generally, checking if a model M satisfies a property $\mathrm{P}_{M}(\Diamond_{x\leq t}\varphi) \geq p$, $p \in [0,1]$ is undecidable [32]. Statistical algorithms are developed in Uppaal SMC for estimating the probability of cost-bounded reachability problems in a given interval of confidence. There are three types of queries: $\mathrm{P}_{M}(\Diamond_{x\leq t}\mathrm{ap})$ (probability estimation), $\mathrm{P}_{M}(\Diamond_{x\leq t}\mathrm{ap}) \geq p$, $p \in [0,1]$ (hypothesis testing), $\mathrm{P}_{M}(\Diamond_{x_1\leq t1}\mathrm{ap}_1) \geq \mathrm{P}_{M}(\Diamond_{x_2\leq t2}\mathrm{ap}_2)$ (probability comparison). In Section 6 we will evaluate the measures of interest for the energy consumption and the probability of failure in Uppaal SMC through probability estimation, while in Section 7 we will compare the approaches based on Möbius and Uppaal SMC. In the next section we will describe the case study.

3 Description of the Case Study

We propose a cyber-physical system from the railway domain as case study: a rail road switch heating system. In this section we briefly describe the real-world devices that we want to model and the underlying logical system we have built for analysis purposes.

3.1 Description of the network of rail road switch heaters

We consider a heating system composed of a series of tubular flat heaters along the rail road track, which warm up the rail road by induction heating. Sensors are used to communicate the temperatures of the air and of the rail road to the rail road switch heating system [46], to perform decisions. The central unit manages the maximum amount of power that can be delivered to the system, in order to prevent possible blackouts. In case of extremely cold conditions, the total amount of energy available may not be sufficient to heat the overall system of switches, hence it is important to duly choose the heaters that must be primarily turned on and those that may be later on. Indeed, in a railway station there are tracks which are less important than others, for example the side tracks, and the heating phase can be delayed for them if necessary. If the temperature cannot be kept above the freezing thresholds, the corresponding switch will experience a failure.

3.2 Logical structure of the system

The two main logical components of our system are the *heater* and the *central coordinator*. The network of heaters is realised by replicating the heater component, and their activation/deactivation is controlled by the central coordinator. In the following we discuss the two main components.

Heater We based the policy employed to activate/deactivate the heating on two threshold temperatures:

- *warning threshold* (T_{wa}): this temperature represents the lowest temperature that the track should not exceed. If the temperature is lowest than T_{wa}, then the risk of ice or snow can lead to a failure of the rail road switch and therefore the heating system needs to be activated;
- *working threshold* (T_{wo}): this is the working temperature of the heating system. Once this temperature is reached, the heating system can be safely turned off in order to avoid an excessive waste of energy.

The energy consumption of the overall system depends on the value of T_{wa} and T_{wo}. A smaller gap between these thresholds will result in a frequent activation of the heating system, but for a shorter period of time. Alternatively, a wider gap between the thresholds will result in a less frequent activation, but it will be for longer periods of time. The time during which a single heater is active depends also on the weather conditions.

Coordinator The coordinator will collect the requests of activation from the pending heaters, and it will manage the energy supply according to a FIFO prioritised order. Indeed, the first heater that asks to be turned on will be the first to be activated. We will assign priorities to switches based on their criticality on the track and we will exploit the assigned priority in the performed analysis, so to guarantee higher reliability to those switches that are essential for the correct functioning of the overall station. The maximum amount of energy deliverable by the system that cannot be exceeded is represented by NH_{max}, and it is measured as the percentage of heaters that can be turned on at the same time. If there is no energy available, each request will be enqueued in the queue of pending heaters. Below we describe in details the behaviour of the central coordinator.

Interactions We now give details about the protocol of communication between the network of heaters and the central coordinator:

– *Heater*: at starting time each heater h_i is switched off and its internal temperature is set to T_{wo}. Once its internal temperature goes below T_{wa}, h_i asks the coordinator to be turned on and waits. Upon reception of the notification, h_i is turned on. After that, two events can happen:
  - the heater h_i reaches an internal temperature above T_{wo}, communicates to the central coordinator the termination of the heating phase and is switched off;
  - a second component h_j with a higher priority asks to be turned on. The energy delivered to h_i is turned off, even though it has not yet reached an internal temperature above T_{wo}. If the temperature is below T_{wa}, h_i will issue a new request of activation to the coordinator.

– *Coordinator*: at starting time the central coordinator is waiting for a message from one of the heaters h_i in the network. Two messages can be received:
  - h_i asks to be activated. This request is inserted in the queue of pending requests in case there is no energy available and the priority of h_i is not higher than that of the already activated switches. Otherwise, the request is accepted and we have two cases; (i) if there is energy available, h_i will be activated by issuing a notification; (ii) if no energy is available but h_i has a priority higher than one of the activated heaters, firstly the heater with lowest priority will be turned off with a notification, and then the activation is notified to h_i;
  - h_i asks to be deactivated. After the deactivation, if there are no heaters that are waiting for being activated then no action is performed. Otherwise, one of the pending heaters h_j (the first in the prioritized queue of pending heaters) is activated by issuing a notification to it.

Before presenting the different models of the system described above, in the next section we will discuss the generic guidelines for modelling energy-saving cyber-physical systems, that we have followed for our case study.

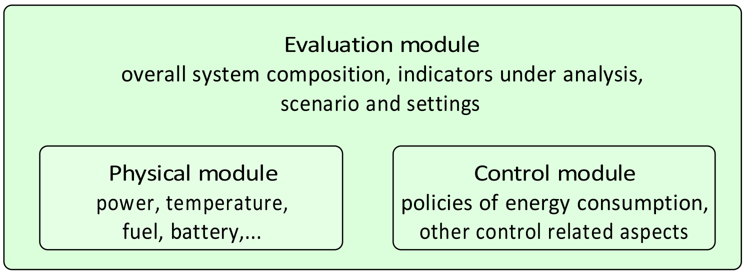

Evaluation module
overall system composition, indicators under analysis,
scenario and settings
Physical module
power, temperature,
fuel, battery,...
Control module
policies of energy consumption,
other control related aspects

Fig.1: The proposed analysis framework for energy saving CPS

4 General Guidelines to Model Energy-saving Cyber-physical Systems

Guidelines on the analysis of reliability and energy consumption indicators of CPS systems are now discussed. Following them, in Section 5 the two analyses approaches are concretely applied to a case study representative of an energy-saving cyber-physical system, to emphasise pros and cons of the two alternatives and to compare the obtained results for cross-validation purposes.

Generally, in energy-saving CPS [41] the supervision of the cyber-control is in charge of strategies to supply energy to components of the physical system, necessary to keep them effective and reliable in the service they accomplish. Our interest is in assessing measures that are representative of the energy consumption, to be combined with other dependability-related properties dictated by the critical domain the CPS is employed in. It is then possible to study the benefits of different energy supply strategies to properly tune the parameters of these strategies toward most rewarding configurations. A diagram of the framework is depicted in Figure 1. The proposed analysis framework is built around three major modules [12].

- *Physical-aspects module*: this module focuses on the physical components of the system and on their characterisation in terms of relevant aspects from the energy viewpoint. It includes models representing phenomena related with energy supply, which depends on the fabric of the supplied components and environmental conditions impacting on the energy consumption. Examples are: i) internal and external temperatures, properly modelled taking into account their evolution in time, given the different means involved (such as iron or copper for the physical components, winter or summer days for the external air); ii) fuel consumption, represented by properly considering the engine parameters, aerodynamic drag, weight, and other relevant parameters; iii) supplied power, represented by properly considering the laws regulating the involved real process; iv) battery charging and others.
- *Control-aspects module*: this module deals with the policies that dynamically regulate the energy consumption of the physical components. To manage

potential complexity while assuring adequate accuracy of the analysis, the representation of such policies is abstracted at the level of their impact on the energy parameters of the controlled physical components. As already mentioned, the primary objective of the proposed model-based approach is to assist the system designer in identifying the best policy to employ among several alternatives, in accordance with pre-established dependability and cost requirements. Trade-offs between energy consumption and dependability requirements are mandatory in critical domains, where, e.g., the energy should not be reduced in safety critical situations. For example, the dynamic power management can be of kind on-off, where energy is supplied or turned off on the basis of values assumed by parameters that depend on physical conditions of the system and of the environment.

- *Evaluation module:* Finally, the third module deals with the composition of the several models from the previous two modules, to end up with the overall evaluation framework. Exercising the overall composed model, energy supply policies trading energy saving and dependability properties can be quantitatively evaluated and compared in terms of properly defined indicators.

The generality of the above outlined approach allows assessing a variety of measures of interest to final customers, service providers and operators, in accordance with the specific application domain where the CPS system under analysis is utilised. Given the aim of trading energy consumption with dependability, typical indicators are energy supplied to individual system components or to the overall system in a certain time interval, as well as failure probability of an individual component or of the overall system due to lack of supplied energy.

A benefit of the proposed approach in Figure 1 is its modularity and compositionality, that allows to be applied to a wide variety of scenarios relevant to CPS, and it has been adopted for our case study. In Section 5 and Section 6 an instantiation of the discussed guidelines is presented, following two different approaches based on SAN and SHA models.

5 Description of the Models

In this section we will describe the models of the rail road switch heating system. We will follow the general guidelines discussed in Section 4. Firstly we will discuss the physical-aspects module of the system under analysis, then we will discuss the controlaspects module, that will be modelled both with SAN models and SHA models. Finally, the evaluation module will be discussed in Section 6.

The models are parameterized based on the two temperature thresholds T_{wa}, T_{wo} and NH_{max} that we recall to be the maximum power that the system can provide at every instant of time, expressed in percentage of heaters that can be turned on at the same time.

5.1 Physical-aspects module

The continuous physical behaviour concerning the increment and decrement of the temperature of the rail road track, respectively when the heater is turned on or off, is modelled by an ODE representing the balance of energy [9].

The heater is represented by a resistance that passes through the rail road in different points in order to warm up the iron. The set-up for the heating device is based on patents of heating switches [17], which contain data about the power consumed by a single heater and about the increment of the temperature of the track in cold winter nights. We assume that the power used by the heater is constant, in order to estimate the kilowatt per hours consumed during the time interval that we consider.

Assuming that the values of the temperature of the surrounding area T_e and the previous internal temperature T are known, the internal temperature T after time t is (we adopt the Newton's notation for differentiation) $T' = (-uA(T-T_e)+Q)/mc$, where u is the coefficient of convective exchange; c, the heat capacity of iron; A, the surface area exposed to the external temperature; m, the mass of the iron bar; Q, the power used when the heater is turned on, if the heater is turned off this value will be zero. We now discuss the stochastic aspects that have been considered as part of the physical module.

Stochastic aspects Stochastic aspects concern the possibility of experiencing a failure in a switch, that will be used for measuring the reliability of the system, and the influence of the weather forecast.

Switch Failures When the temperature of the rail road track is below the freezing threshold (i.e. 0°C in our experiments), a switch may experience a failure. In this case, the time-to-failure is modelled with an exponential distribution with fixed rate, that will be based on the temperature of the rail road track.

Weather forecast To model the external weather conditions, our model takes in input data structures containing profiles of average temperatures in those days for which the analysis is relevant (e.g. winter days), as depicted in Figure 2. For our experiments, we have five different daily weather profiles retrieved from the internet [54]. The time window under analysis is divided into intervals to which an average reference temperature is assigned. The current instance of the model concentrates on a whole day, divided into intervals of two hours. However, the model can be easily modified to consider longer (or shorter) periods, as well as different number of intervals. A probability is assigned to each weather profile, based on weather statistics.

5.2 Control-aspects module

The control-aspects module has been modelled both in SAN and SHA models. Comparisons between these two different approaches will be derived in Section 7.

SAN Models We now describe the SAN based model realising the rail road switch heating system logic and the protocol described in Section 3, built through the functionalities provided by Möbius.

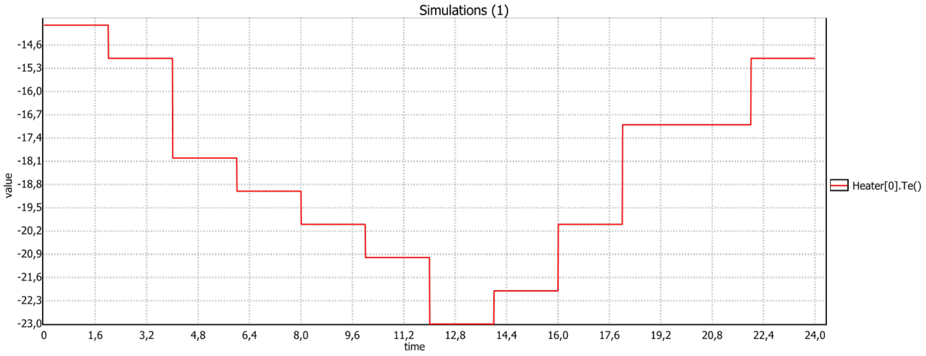


Fig.2: A weather profile corresponding to the temperatures for coldest winter nights in a northern city, retrieved from [54]. The simulation starts at 6:pm and terminates after 24 hours.

The overall model is obtained by the composition of the atomic models, using the join and rep operators, as shown in Figure 3 (where the atomic models are the leaves of the tree while the overall composed model is the root). The atomic model *Coordinator* represents the central coordinator and it will interact with the network of switch heaters. The atomic SAN models *LocalitySelector*, *ProfileSelector* and *SwitchIDSelector* represent, respectively, the selector for the weather profile, the location of the switch and the unique identifier of each switch. The submodel *HeaterModuleM* represents an instance of a single heater module, obtained by the composition, using the join operator, of the four atomic SAN models. Those atomic models share the places relative to the locality of the device, its weather profile and the unique ID. The submodel *HeatersNetM*, obtained by replicating *numRep* times the model *HeaterModuleM* using the Rep operator, represents the network of heaters, where the parameter *numRep* identifies the number of devices composing the network. Finally, the model *SwitchHeatingSysM*, obtained using the join operator, represents the overall system.

All these SAN models interact through *shared places*, a feature available in Möbius [18] for joining different SAN models thus allowing modularity.

Note that in Möbius different replicas of SAN models are anonymous, and it is not possible to distinguish between them. Hence, the *SwitchIDSelector* SAN model is used for assigning a unique ID to each heater module.

The main SAN model concerning the rail road switch heater is depicted in Figure 4. It is partitioned into three logical components: the *init sub-net*, the *clock sub-net* and the *heater sub-net*. The *init* subnet initialises the data structures used by the SAN model. The clock sub-net models the evolution of time (during one day in our analyses), and it is used to update the environment temperature and the temperature of the rail road track. In this paper we have considered as unit of time one hour. The activity *clock* has a deterministic distribution of time (non Markovian), and completes each hour. When *clock* completes, the place *Temperature* is updated: if the heater is turned on then the temperature increases, otherwise the temperature will be updated according to the temperature of the environment, as in the equation in Section 5.1.

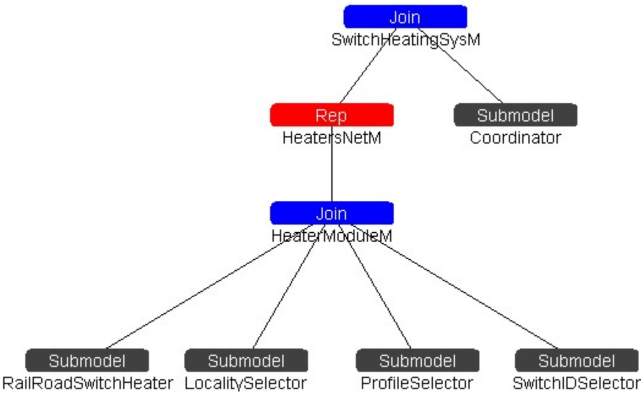


Fig.3: The composed model.

The heater sub-net represents the status of the rail road switch heater. The heater can be activated (one token in the place *on*), waiting for being activated (one token in the place *ready*), turned off (one token in the place *off*), or failed (one token in the place *failure*). Indeed, according to the heating policy, once the system temperature falls below a pre-defined warning threshold (T_{wa}), the heating needs to be activated, otherwise the associated switch fails. Then, once the temperature raises and reaches the working threshold (T_{wo}), the heating system can be safely turned off.

The heater sub-net interacts with the *Coordinator* SAN model through places shared among all the replicas of the heater model and the *Coordinator* model. For example, if the heater is in state ready, in order to be turned on, the input gate $i1_{ready2on}$ checks if the marking of the shared place *notifyIn* is equal to the marking of the place *SwitchID*, which means that the coordinator has notified the heater to be turned on.

The function representing the heating exchange is defined in C++, and it is called by the output gate $O1_{clock}$ in Figure 4 to update the temperature of the rail road each interval of time *t*. The activity $TA_{failure}$ models the failure of a component. It has an exponential distribution of time based on the temperature of the rail road track: the more the temperature is below the freezing threshold the more is probable that the activity will fire (the activity is not activated if the temperature is positive).

The SAN model Coordinator represents the central management unit and it interacts with all the heaters in the network by activating, deactivating or moving them in a waiting state. Below we will describe the network of stochastic hybrid automaton used for modelling the system of rail road switch heaters.

SHA Models The rail road switch heating system has also been modelled through stochastic hybrid automata, with the purpose of cross-validating the two distinct formalisations and improve the trustworthiness of the obtained results.

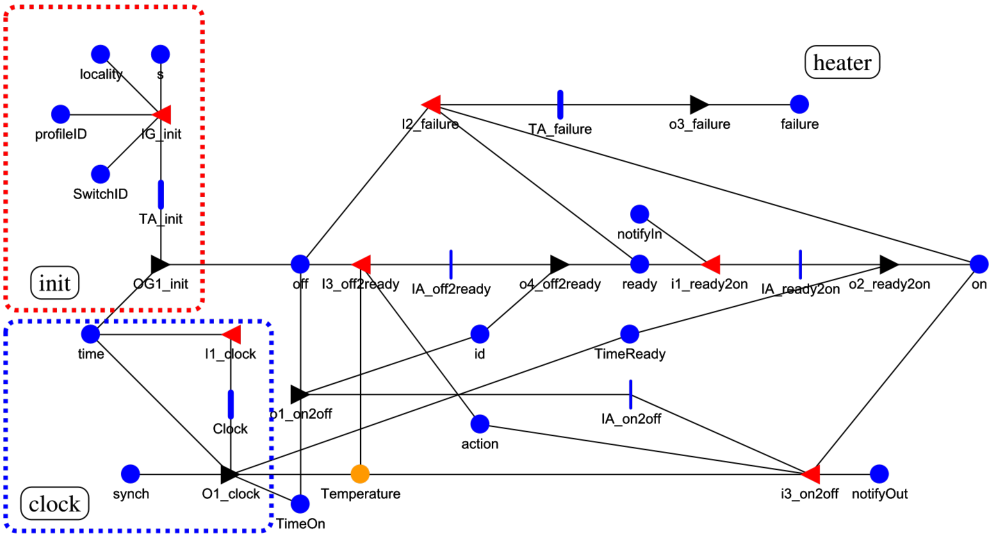


Fig.4: The SAN model RailRoadSwitchHeater, logically divided into three sub-nets: the init sub-net, the clock sub-net and the heater sub-net.

Indeed, this formalism allows to capture both discrete, continuous and stochastic aspects in a single framework. We have been able to verify the correctness of the interactions, as well as energy and reliability indicators, by using the Uppaal SMC toolkit. We briefly outline the formalisation of the system of (remotely controlled) rail road switch heaters as a product of hybrid automata.

The ODE in Section 5.1 is expressed in the stochastic hybrid model H in Figure 5, where the temperature *T* is a *continuous clock* and the flow function *F* (i.e. the ODE) is similar in different states. Indeed, when H is in state on, *F* is adding the term Q (i.e. the power); this is not the case in states off and ready.

The two main logical components describing the discrete cyber part of the analysed system are the *heater* H (depicted in Figure 5) and the *central coordinator* K (depicted in Figure 6). The network composed of *n* heaters and the coordinator is realised by making the product of K and the replicas of the stochastic hybrid automaton H_{id}, *id* ∈ 1,…,*n*, where each heater is uniquely identified by its *id*, i.e. $(\bigotimes_{id \in 1,\ldots,n} H_{id}) \otimes K$. The heater model is in Figure 5 and it implements the policy for activating and deactivating the heating phase, similarly to the SAN model in Figure 4. In particular, the dotted transitions are urgent (i.e. instantaneous) probabilistic transitions used for selecting one of the available weather profiles. The main states are on, off, ready and fail, which correspond to the places of the SAN model *RailRoadSwitchHeater*. Here we note that each state has an inner cycle modelling the decrease and increase of the internal temperature according to the flow function, and that both the incoming transitions to state fail have an exponential distribution of time. During a simulation, the current time is stored in the clock x, and a variable hour stores the current hour.

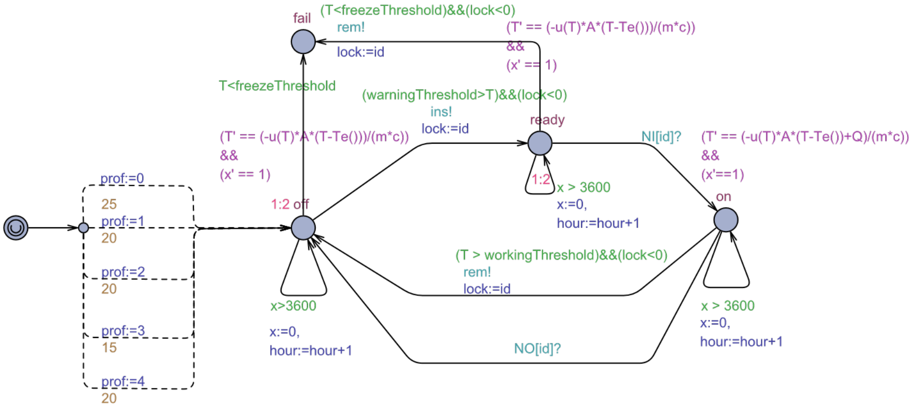


Fig.5: The stochastic hybrid automaton H, modelling an instance of a rail road switch heater

The function Te(), used in the flow function of T, selects the actual external temperature based on the current hour, and it is implemented in Uppaal.

The coordinator is modelled as the hybrid automaton in Figure 6. Its behaviour is similar to the one of the SAN model*Coordinator*. The queue of pending heaters is modelled with the array queue[] of length equal to NH_{max}, and the functions enqueue(id) and dequeue() are used for inserting and removing elements, while empty() returns true if the queue of pending heaters is empty.

The coordinator sends messages to the network of heaters through two arrays of channels NI[id] and NO[id] indexed by the identifiers of the heaters, to notify respectively the activation and deactivation of a heater (see Section 3 for the communication protocol). Note that Uppaal SMC only allows broadcast channels, hence an array of channels has been adopted in order to implement one-to-one communications. Following the standard notation, sending through a channel a is denoted as a!, while reading as a?. Upon reception of the notification NI[id]?, the heater with identifier id switches from state ready to state on.

The heaters communicate to the coordinator their transition from off to ready through the channel ins, so asking for being activated, and their transition from on to off through the channel rem; both channels are many-to-one. All channels are *urgent*, which means that no delays will occur in case a synchronisation is available.

While the coordinator is in a busy state, a shared variable lock is used as a semaphore to prevent a heater from sending messages that cannot be elaborated, and it is used by the heaters for communicating their identifiers to the coordinator. In the next section we will describe the evaluation module for both SAN and SHA models.

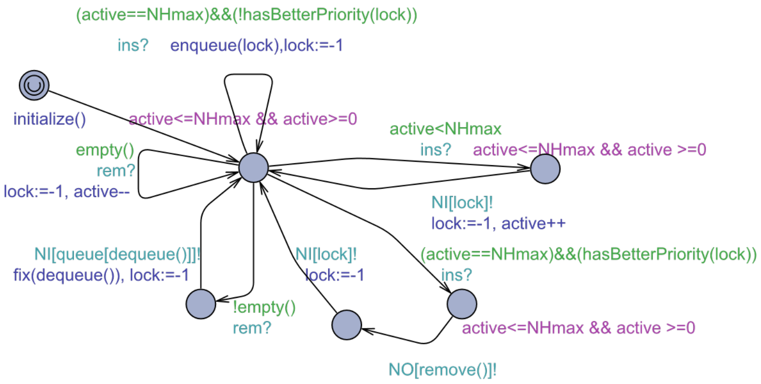

(active==NHmax)&&(!hasBetterPriority(lock))
ins? enqueue(lock),lock:=-1
initialize()
active<=NHmax && active>=0
active<NHmax
ins? active<=NHmax && active >=0
empty()
rem?
lock:=-1, active--
NI[lock]!
lock:=-1, active++
NI[queue[dequeue()]]!
fix(dequeue()), lock:=-1
NI[lock]!
lock:=-1
(active==NHmax)&&(hasBetterPriority(lock))
ins?
!empty()
rem?
active<=NHmax && active >=0
NO[remove()]!

Fig.6: The stochastic hybrid automaton K, modelling the coordinator

6 Evaluation module

In the evaluation module we will evaluate the energy consumption and the reliability of the system considering different thresholds-based policies of energy consumption. Concerning the scenarios and settings we have considered for our analysis, real world data are used. In particular, to keep the presentation simple, the layout of an average small-size railway station has been chosen. The cold winter days are considered for the temperatures and the parameters of the physical model are based on data available from real devices [17] [46].

6.1 Measures of Interest

We consider two different measures of interest. The first concerns the energy consumption while the second addresses the reliability of the system under analysis.

1 $CE(t,l)$: the time (in hours) a generic heater is activated in the time interval $[t,t+l]$. By multiplying $CE(t,l)$ for the power consumed (kilowatt per hour), that is the term Q in the heat exchange equation (see Section 5.1), it is possible to derive the energy consumed by the system;

2 *PFAIL*(*t,l*): the probability that a generic switch fails (becomes frozen) at time *t* +*l*, given that at time *t* is not failed.

We remark that reliability is computed as the probability that no failure occurs in the interval of time under analysis [52], that is 1−*PFAIL*(*t,l*).

Measures of Interest in the SAN models Concerning the SAN models, in Möbius reward structures have been used for evaluating the measures of interest. In particular, *CE*(*t,l*) is defined by accumulating in the interval [*t,t* + *l*] the time that each replica of the SAN model *RailRoadSwitchHeater* spends in markings with one token in place *on* (see Figure 4) , that is the time that each heater is activated. The measure *PFAIL*(*t,l*) is defined as the probability that at time *t* +*l* there is one token in the place *failure* of the SAN model *RailRoadSwitchHeater*.

Measures of Interest in the SHA models In Uppaal SMC, the measures of interest are defined as formulae in MITL, enriched with quantification operators on the replicated models and expected values. We will consider a discrete clock energy that counts the hours H spends in state on. For enhancing readability, we have omitted this clock in Figure 5. For the energy consumption we estimate the number of hours in which the heaters are active as:

$$CE(t,l) = \mathrm{E}[<= 24;10000]\ (\max : \sum_{\mathrm{i:idt}} \mathrm{H_i\ energy})$$

where E stands for the expected value, 24 is the considered interval of time (24h) and 10000 are the simulations executed by the tool. The overall energy consumption is the sum for all H_i of all the clocks energy.

The probability of failure is estimated by Uppaal SMC with the formula:

$$PFAIL(t,l) = \mathrm{P}(\Diamond_{\mathrm{h} \leq 24} \exists(\mathrm{i} : \mathrm{id_t})(\mathrm{H_i fail}))$$

The above formula evaluates the probability that in the interval [*t,t* + *l*] (24h) there exists at least a switch H_i in the network which has failed, i.e. H_i is in state fail.

6.2 Evaluation

We now discuss the evaluation of *CE*(*t,l*) and *PFAIL*(*t,l*) on the two adopted tools. In our experiments we assume to have a network with 10 switches, partitioned into 4 switches with high priority, 3 with medium priority and 3 with low priority. We report the results of the experiments performed with 9 pairs of T_{wa} and T_{wo}, that are those with a better trade-off between energy consumption and probability of failure. The amount of energy available to the system is set to NH_{max}= 50%, the values for T_{wa} are respectively 6°C, 7°C and 8°C, and the values for T_{wo} are 1°C or 2°C higher than the corresponding value of T_{wa}. We have considered cumulative values of *CE*(*t,l*) and *PFAIL*(*t,l*) for both heaters with high, medium and low priorities in the network. In Figure 7 and Figure 8 the results of the experiments are reported.

The measures of interest have been evaluated in Möbius through simulation, with minimum and maximum batches per experiment set to respectively 1000 and 10000 and minimum interval of confidence level of 0.95. Concerning Uppaal SMC, the number of simulations used for evaluating *CE*(*t,l*) has been fixed to 10000, while for *PFAIL*(*t,l*) from a minimum of 800 to a maximum of 10000 simulations have been performed, that are those necessary for reaching a confidence interval of 0.995.

The estimated values of *CE*(*t,l*) are displayed in Figure 7a and Figure 8a, and as expected both tools report similar results. In particular, by augmenting the value of T_{wa} the system consumes more energy for reaching a higher temperature. On the converse, we note that by augmenting the difference T_{wo}- T_{wa}, the values of *CE*(*t,l*) slightly decrease (with the only exception of T_{wa}=6°C). Indeed, when augmenting T_{wo}-T_{wa}, the probability of failure increases, and the failed switches will no longer consume energy, resulting in an overall decrease of *CE*(*t,l*).

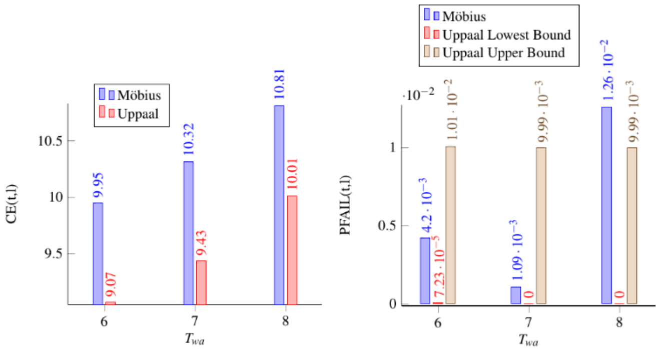
In Figure 7b and Figure 8b the values of *PFAIL*(*t,l*) are reported. Note that in this case Uppaal SMC only reports the intervals of the probability estimation. Also in this case the tools report similar results, with the main difference being the case T_{wa}=8°C and T_{wo}=10°C, which is the worst (i.e. highest) probability of failure. In particular, both tools identify as optimal pair of thresholds the values T_{wa}=7°C and T_{wo}=8°C. Indeed in this case we have the lowest probability of failure, that is a better distribution of energy among all the heaters with different priorities. This is because with a tighter gap of $T_{wo}T_{wa}$=1°C the time needed for reaching T_{wo} is less than for the case of T_{wo}- T_{wa}=2°C. Concerning the performance of the experiments, it has been used a machine with CPU Intel Core i5-4570 at 3.20 GHZ with 8 GB of RAM, running 64-bit Windows 10, Uppaal SMC academic version 4.1.19 (rev. 5649) and Möbius 2.5.0. The elapsed running time is reported in Table 1. The longest running time in Möbius has been of 11.049 sec. for computing both measures of interest with parameters T_{wa}=6°C and T_{wo}=7°C. On the converse, Uppaal SMC has shown worst performances. Indeed, with parameters T_{wa}=6°C and T_{wo}=7°C, we had a running time 295.922 sec. for *CE*(*t,l*) and 30.39 sec. for *PFAIL*(*t,l*).

Table 1: The elapsed running time of the experiments for different settings of (T_{wa},T_{wo}): for Möbius the total running time of each experiment is reported (i.e. both measures), while for Uppaal SMC the time needed for computing each measure is reported

	Uppaal (*PFAIL*(*t,l*))	Uppaal (*CE*(*t,l*))	Möbius Total Running Time
(6°C,7°C)	30.39s	295.922s	9.657s
(7°C,8°C)	24.922s	313.968s	9.405s
(8°C,9°C)	18.422s	316.344s	3.868s
(6°C,8°C)	233.516s	293.875s	5.387s
(7°C,9°C)	25.015s	294.797s	2.093s
(8°C,10°C)	259.219s	290.704s	1.775s

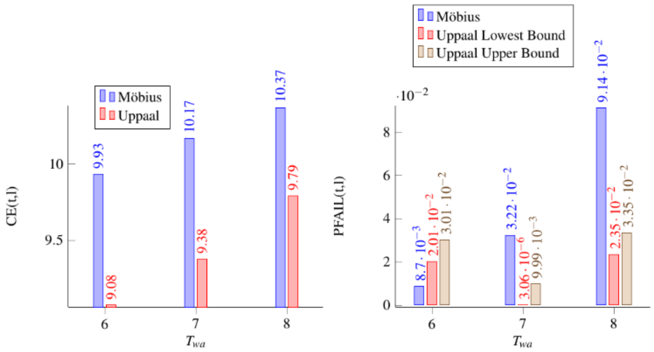
However, we note that for evaluating *CE*(*t*,*l*) Uppaal SMC is forced to perform 10000 simulations, and for *PFAIL*(*t*,*l*) we have a tight confidence level, i.e. 0.995. Nevertheless, in this case study the performances of Möbius overwhelm those of Uppaal SMC.

Finally, with both approaches it has been possible to model and evaluate the case study, and to cross-validate the obtained results.



(a) The energy consumption (b) The probability of failure

Fig.7: The measures of interests with T_{wa}- T_{wo}= 1°C and NH_{max}= 50%



(a) The energy consumption (b) The probability of failure

Fig.8: The measures of interests with T_{wa}- T_{wo}= 2°C and NH_{max}= 50%

8 Related Work

There is a wide literature concerning the analysis and optimisation of energy-saving systems in several application domains using formal approaches; in the following we discuss some of these recent efforts.

An example of combining stochastic simulation and model checking is in [44], where a tool chain comprising Uppaal and Möbius is used for the proactive schedule generation for manufacturing scenarios with resource competition, stochastic resources breakdowns, and earliness/tardiness penalties. The system is modelled with Modest [29], the optimal schedule is synthesised with the tool Uppaal Cora [40,22] and simulations carried on with Möbius. Optimising energy consumption for energy aware buildings, represented as stochastic hybrid automata, is the selected case study in [20]. Statistical Model Checking and analysis of variance has been used to identify Pareto-optimal configurations in terms of both discomfort and energy consumption. In our analysis, we identify the best trade-off between energy consumption and reliability through simulations. An approach for estimating the energy consumption of mobile apps is proposed in [43]. Similarly, we build an abstract model to predict the energy consumption of the rail road switch heating system at the variation of temperature thresholds and available energy, to find the optimal setup of the parameters. Dynamic Power Management (DPV) and Dynamic Voltage and Frequency Scaling (DVFS) are adopted in [1] by using Statistical Model Checking. The system is modelled as Stochastic Hybrid Games. The tool Uppaal Stratego [21] is used to synthesise the safe and near optimal strategy. Similarly, we adopted Statistical Model Checking and DPM, to turn off the energy consumption of heaters when a given temperature is reached. Hybrid automata have been used in [26] to study the dynamic power management control problem. We do not consider a power adjustment mechanism in the rail road switch heating system. A prediction mechanism for minimising the power supplied could be added for example in case of warmer nights.

The applicability of self-organizing systems for different fields of power system control is discussed in [45]. In our case we consider a central unit which manages the different heaters. The demand of energy is adjusted according to the maximum energy that can be delivered by the central unit. In case of failure of a heater, the energy is automatically shared among the remaining active heaters. We show that by managing the temperature thresholds it is possible to improve reliability even in case of low energy demand. The survivability of a smart house is analysed in [27]. Hybrid Petri Nets [23] are used for modelling this scenario. The authors consider a randomly chosen probability of failure and fixed thresholds, while in our case the probability of failure is derived from the model and we instantiate the thresholds to improve the energy consumption and reliability. The trade-off between energy saving and reliability is studied in [55], by managing frequencies and voltage of the delivered energy. In our approach different policies of energy consumption are based on thresholds temperatures, and we do not deal with frequencies and voltage of the delivered energy and assume a fixed amount of power. Services negotiation of energy and reliability requirements is the selected case study in [53]. The entities and their protocol of communication have been modelled with Remes Hdcl language [50] and rendered as timed automata [3], through which the absence of deadlock has been certified with Uppaal. Their reliability and energy requirements are given parameters that are negotiated between the parties; instead in our approach we estimate those values based on policies of energy consumption.

9 Conclusion

We have addressed cross-validation of reliability and energy consumption of a critical cyber-physical system belonging to the railway domain (a rail road switch heating system), by comparing the approach based on Stochastic Activity Network and Möbius and the approach based on Stochastic Hybrid Automata and Uppaal SMC.

We address some lines of future research concerning energy saving CPS. The behaviour of CPS is in general unpredictable, because of physic and environmental aspects that are involved in their design. A future line of research concerns the introduction of control techniques for restricting their possible behaviours in order to predict and avoid possible failures, improve and verify their dependability [4]. It is also interesting to study the formal specification of the energy requirements that a CPS must satisfy, verifying that the proposed model satisfies them or proving that such requirements are not satisfiable [8].

References

1. Ahmad, W., v. d. Pol, J.: Synthesizing energy-optimal controllers for multiprocessor dataflow applications with uppaal stratego. In: ISOLA 2016 (2016)
2. Ahmed, W., Hasan, O., Tahar, S.: Formal dependability modeling and analysis: A survey. In: CICM 2016, Bialystok, Poland, July 25-29. LNAI, vol. 9791, pp. 132–147. Springer (2016)
3. Alur, R., Dill, D.L.: A theory of timed automata. TCS 1994 126(2), 183 – 235
4. Antsaklis, P.: Goals and challenges in cyber-physical systems research editorial of the editor in chief. IEEE Transactions on Automatic Control 59(12), 3117–3119 (Dec 2014)
5. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. Dependable and Secure Computing, IEEE Transactions on 1(1), 2014
6. Baier, C., Haverkort, B.R., Hermanns, H., Katoen, J.P.: Performance evaluation and model checking join forces. Commun. ACM 53(9), 76–85 (Sep 2010)
7. Balbo, G.: Introduction to generalized stochastic Petri nets. In: Bernardo, M., Hillston, J. (eds.) Formal Methods for Performance Evaluation, LNCS, vol. 4486. Springer (2007)
8. Banerjee, A., Venkatasubramanian, K.K., Mukherjee, T., Gupta, S.K.S.: Ensuring safety, security, and sustainability of mission-critical cyber-physical systems. Proceedings of the IEEE 100(1), 2012
9. Basile, D., Chiaradonna, S., Di Giandomenico, F., Gnesi, S.: A stochastic model-based approach to analyse reliable energy-saving rail road switch heating systems. JRTPM (2016), http://www.sciencedirect.com/science/article/pii/S2210970616300051
10. Basile, D., Chiaradonna, S., Di Giandomenico, F., Gnesi, S., Mazzanti, F.: Stochastic model based analysis of energy consumption in a rail road switch heating system. In: SERENE 2015. LNCS 9274
11. Basile, D., Di Giandomenico, F., Gnesi, S.: Tuning energy consumption strategies in the railway domain: a model-based approach. In: 7TH International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, ISOLA 2016 (2016)
12. Basile, D., Di Giandomenico, F., Gnesi, S.: Model-based evaluation of energy saving systems. In: Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (eds.) Green IT Engineering: Concepts, Models, Complex Systems Architectures. Studies in Systems, Decision and Control, vol. 74, pp. 187–208. Springer International Publishing, Cham (2017)

13. Basile, D., Giandomenico, F.D., Gnesi, S.: Enhancing models correctness through formal verification: A case study from the railway domain. In: MODELSWARD 2017, Porto, Portugal, February 19-21, 2017. pp. 679–686 (2017)
14. Basile, D., Giandomenico, F.D., Gnesi, S.: Statistical model checking of an energy-saving cyber-physical system in the railway domain. In: Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017. pp. 1356–1363 (2017)
15. Bause, F., Kritzinger, P.S.: Stochastic Petri nets: An introduction to the theory. SIGMETRICS Perform. Eval. Rev. 26(2)
16. Bernardi, S., Merseguer, J., Petriu, D.C.: Model-Driven Dependability Assessment of Software Systems. Springer (2013)
17. Brodowski, D., Komosa, K.: A railroad switch and a method of melting snow and ice in railroad switches (2013), https://data.epo.org/publication-server/rest/v1.0/ publication-dates/20131225/patents/EP2677079NWA1/document.html
18. Clark, G., Courtney, T., Daly, D., Deavours, D., Derisavi, S., Doyle, J.M., Sanders, W.H., Webster, P.: The Möbius modeling tool. In: PNPM. pp. 241–250 (2001)
19. Clarke, Jr., E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT Press, Cambridge, (1999)
20. David, A., Du, D., Guldstrand Larsen, K., Legay, A., Mikucionis, M.: NFM 2013, chap. Op- timizing Control Strategy Using Statistical Model Checking, pp. 352–367. Springer (2013)
21. David, A., Jensen, P.G., Larsen, K.G., Mikucionis, M., Taankvist, J.H.: Uppaal Stratego, pp. 206–211. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
22. David, A., Larsen, K.G., Legay, A., Mikuaionis, M., Poulsen, D.B.: Uppaal smc tutorial. Int.ˇ J. Softw. Tools Technol. Transf. 17 (2015)
23. David, R., Alla, H.: On hybrid Petri nets. DEDS 11(1-2), 9–40 (2001)
24. Diab, Hassan B.; Zomaya, A.Y.: Dependable computing systems: paradigms, performance issues and applications. John Wiley & Sons (2005)
25. Doukas, N.: Technologies for greener internet of things systems. In: Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (eds.) Green IT Engineering: Components, Networks and Systems Implementation. Studies in Systems, Decision and Control, vol. 105, pp. 23–42. Springer International Publishing, Cham (2017)
26. Erbes, T., Shukla, S.K., Kachroo, P.: Stochastic learning feedback hybrid automata for dynamic power management in embedded systems. In: SMCia/05 (2005)
27. Ghasemieh, H., Haverkort, B.R., Jongerden, M.R., Remke, A.: Energy resilience modeling for smart houses. In: 45th Annual IEEE/IFIP, DSN 2015. pp. 275–286. IEEE C.S. (2015)
28. Hahanov, V., Litvinova, E., Chumachenko, S.: Green cyber-physical computing as sustainable development model. In: Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (eds.) Green IT Engineering: Components, Networks and Systems Implementation. Studies in Systems, Decision and Control, vol. 105, pp. 65–85. Springer International Publishing, Cham (2017)
29. Hartmanns, A., Hermanns, H.: The Modest Toolset: An Integrated Environment for Quantitative Modelling and Verification, pp. 593–598. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
30. Haverkort, B.R.: Lectures on formal methods and performance analysis. chap. Markovian Models for Performance and Dependability Evaluation, pp. 38–83. Springer-Verlag (2002)
31. Henzinger, T.A.: The theory of hybrid automata. pp. 278–. LICS ’96, IEEE C.S. (1996)
32. Henzinger, T.A., Ho, P.: Algorithmic analysis of nonlinear hybrid systems. In: CAV (1995)
33. Hinton, A., Kwiatkowska, M., Norman, G., Parker, D.: Prism: A tool for automatic verification of probabilistic systems. In: TACAS 2006, volume 3920 of LNCS. pp. 441–444. Springer
34. Kemper, P., Tepper, C.: Traviando - debugging simulation traces with message sequence charts. Quantitative Evaluation of Systems, International Conference on 0, 135–136 (2006)
35. Kharchenko, V., Illiashenko, O.: Concepts of green it engineering: Taxonomy, principles and implementation. In: Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (eds.) Green IT Engineering:

Concepts, Models, Complex Systems Architectures. Studies in Systems, Decision and Control, vol. 74, pp. 3–19. Springer International Publishing, Cham (2017)

36. Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (eds.): Green IT Engineering: Components, Networks and Systems Implementation, Studies in Systems, Decision and Control, vol. 105. Springer Publishing Company, Incorporated, Cham, 1st edn. (2017)
37. Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (eds.): Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control, vol. 74. Springer Publishing Company, Incorporated, Cham, 1st edn. (2017)
38. Krishna, S.N., Trivedi, A.: Hybrid automata for formal modeling and verification of cyberphysical systems. CoRR (2015), http://arxiv.org/abs/1503.04928
39. Larsen, K.G., Legay, A.: Statistical model checking: Past, present, and future. In: Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part I. pp. 3–15 (2016)
40. Larsen, K.G., Pettersson, P., Yi, W.: Uppaal in a nutshell. JSTTT 1 (1997)
41. Lee, E.A.: Cyber physical systems: Design challenges. ISORC '08, IEEE C.S. (2008)
42. Legay, A., Delahaye, B., Bensalem, S.: RV 2010. Proceedings, chap. Statistical Model Checking: An Overview. Springer
43. Lu, Q., Wu, T., Yan, J., Yan, J., Ma, F., Zhang, F.: Lightweight method-level energy consumption estimation for android applications. In: 10th International Symposium on Theoretical Aspects of Software Engineering, TASE 2016, Shanghai, China, July 17-19, 2016. pp. 144–151. IEEE (2016), http://doi.ieeecomputersociety.org/10.1109/TASE.2016.27
44. Mader, A., Bohnenkamp, H., Usenko, Y.S., Jansen, D.N., Hurink, J., Hermanns, H.: Synthesis and stochastic assessment of cost-optimal schedules. International Journal on Software Tools for Technology Transfer (STTT) 12(5), 305–317 (November 2009), http://doc.utwente.nl/69344/
45. Muller, S.C., Hager, U., Rehtanz, C., Wedde, H.F.: Application of self-organizing systems in power systems control. In: Dieste, O., Jedlitschka, A., Juzgado, N.J. (eds.) PROFES 2012 Proceedings. LNCS, vol. 7343, pp. 320–334. Springer (2012)
46. http://www.railsco.com/electric_switch_heater_controls.htm, (accessed on June 2016)
47. Reibman, A., Smith, R., Trivedi, K.: Markov and Markov reward model transient analysis: An overview of numerical approaches. European Journal of Operational Research 40(2) (1989)
48. Sanders, W.H., Meyer, J.F.: Stochastic activity networks: Formal definitions and concepts. In: Lectures on Formal Methods and Performance Analysis (2000)
49. Sanders, W.H., Nicol, D.M., Trivedi, K.S.: Model-based evaluation: From dependability to security. IEEE Transactions on Dependable and Secure Computing 1(undefined), 48–65 (2004)
50. Seceleanu, C., Vulgarakis, A., Pettersson, P.: Remes: A resource model for embedded systems. In: Engineering of Complex Computer Systems, 2009. pp. 84-94 (2009)
51. Front matter. In: Karlin, H.M.T. (ed.) An Introduction to Stochastic Modeling (Revised Edition), Academic Press, revised edition edn. (1994), http://www.sciencedirect.com/science/article/pii/B978012684885450001X
52. Trivedi, K.S.: Probability & statistics with reliability, queuing and computer science applications. John Wiley & Sons (2008)
53. Causevic, A., Seceleanu, C., Pettersson, P.: Distributed energy management case study: A formal approach to analyzing utility functions. In: ISoLA Proc.II, LNCS, vol. 8803, pp. 74–87. Springer Berlin Heidelberg (2014)
54. https://weatherspark.com/#!graphs;ws=27985, (accessed on March 2016)
55. Zhu, D., Melhem, R., Mossè, D.: The effects of energy management on reliability in real-time embed ded systems. In: ICCAD. pp. 35–40 (Nov 2004)