30 Years of Simulation-Based Quantitative Analysis Tools: a Comparison Experiment between Möbius and Uppaal SMC

Davide Basile[1] **Maurice H. ter Beek**[1] Felicita Di Giandomenico[1]
Alessandro Fantechi[1,2] Stefania Gnesi[1] Giorgio O. Spagnolo[1]

[1] ISTI–CNR, Pisa
[2] University of Florence

ISoLA 2021

28 October 2021

30 years of cross-fertilisation . . .

. . . between the fields of *performance evaluation* and *formal verification*

Baier, Haverkort, Hermanns & Katoen, *Performance Evaluation and Model Checking Join Forces* @ CACM 2010

- Stochastic model-based analysis has a longstanding and rich history in Mathematics, well preceding Computer Science as a discipline
- Möbius can be traced back to its predecessors UltraSAN & MetaSAN

Sanders & Meyer, *METASAN: A Performability Evaluation Tool Based on Stochastic Activity Networks* @ 1986 Fall Joint Computer Conference

- SMC roots: hypothesis testing in context of probabilistic bisimulation

Larsen & Skou, *Bisimulation through Probabilistic Testing* @ Inf. Comput. 1991

- SMC tools are very recent: 1st release of Uppaal SMC in 2014

Agha & Palmskog, *A Survey of Statistical Model Checking* @ ACM Trans. Model. Comput. Simul. 2018

Contribution

A *comparison experiment* on selected features between two frameworks:

- Stochastic Activity Networks (SAN) and Möbius

  Basile, Chiaradonna, Di Giandomenico & Gnesi, *A stochastic model-based approach to analyse reliable energy-saving rail road switch heating systems* @ J. Rail Transp. Plan. Manag. 2016

- Stochastic Hybrid Automata (SHA) and Uppaal SMC

  Basile, Di Giandomenico & Gnesi, *Statistical Model Checking of an Energy-Saving Cyber-Physical System in the Railway Domain* @ SAC 2017

Case studies from our industry partners in the regional project STINGRAY (SmarT station INtelliGent RAilwaY):

- station lighting: whenever (time) and wherever (space) possible reduce illumination, guaranteeing minimum levels as requested by legislation

  ter Beek, Ciancia, Latella, Massink & Spagnolo, *Spatial Model Checking for Smart Stations @ FMICS 2021*

- *heating of railroad switches in ice conditions*

Case study: railroad switch heating system

- **Rail road switches**: critical system
- **Heating system**: tubular flat heaters, induction heating, possibility for energy management
- **Cyber-Physical System**: sensors for the temperatures
  - weather forecast: *stochastic* (physical)
  - temperature evolution: *continuous* (physical)
  - on/off power management: *discrete* (cyber)
- **Quantitative properties**: energy vs. reliability through policies of energy consumption

Energy-saving policies

Dynamic Power Management, on/off policy based on two thresholds:

- on warning threshold T_{wa} guarantees reliability
- off working threshold T_{wo} guarantees energy saving

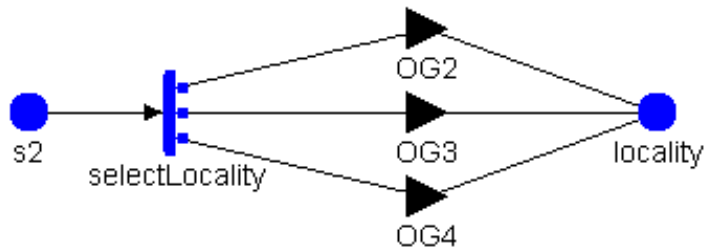
3rd parameter NH_{max}: maximum power supply (% of heaters turned on)

FIFO priorities for main railroad tracks

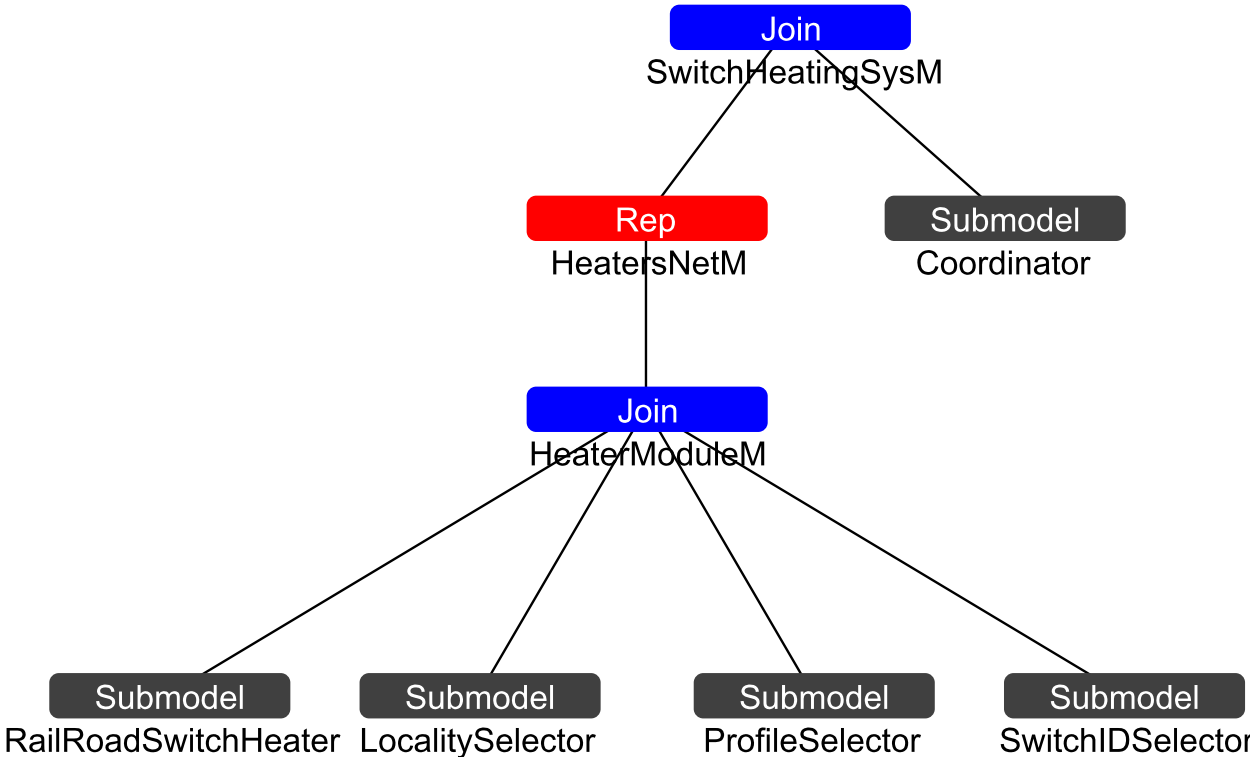
The continuous physical behaviour concerning temperature increment and decrement of the railroad track when the heater is turned on or off, resp., is modelled by an ODE (representing the energy balance)

Stochastic model-based analysis and Möbius

- **Möbius**: distributed discrete-event simulator and explicit state-space generators and numerical solution algorithms for Markovian models; analysis of both transient and steady-state reward models
- **Stochastic Activity Networks**: generalisation of stochastic Petri nets
  - places and activities: same interpretation as places and transitions in PNs
  - 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 can be *instantaneous* or *timed*
    - instantaneous activities complete once enabling conditions are satisfied
    - timed activities take time to complete (stochastic distribution of time)
  - cases are associated to activities and are used to represent probabilistic uncertainty about the action taken upon completion of the activity
  - policies of activation/reactivation of activities
  - primitives are defined using C++ code

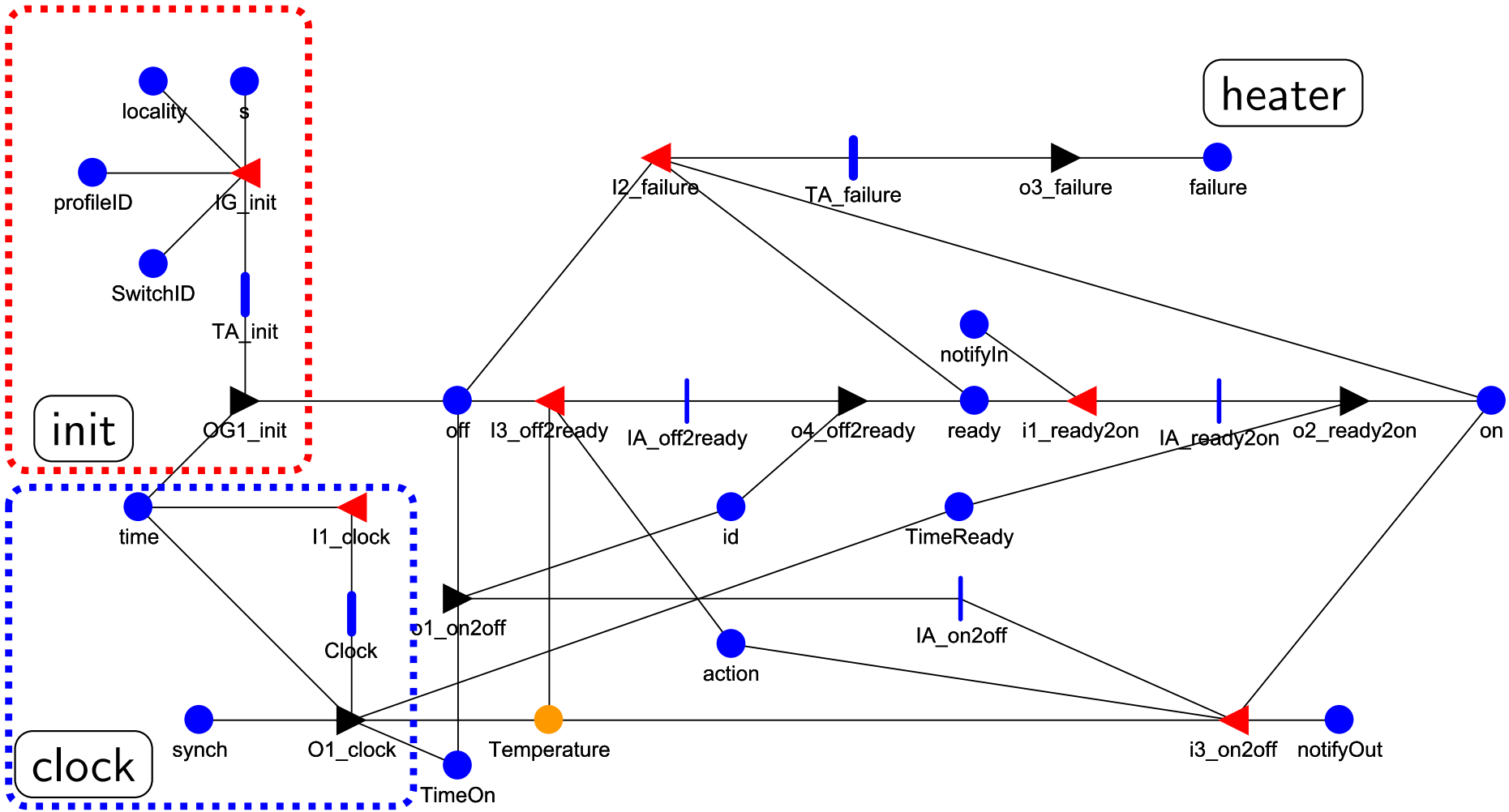


SAN Composed Model



Interactions through shared places

RailRoadSwitchHeater SAN



$$mc\frac{\partial T}{\partial t} = -uA(T - T_{env}) + \dot{Q}$$

Statistical Model Checking and Uppaal SMC

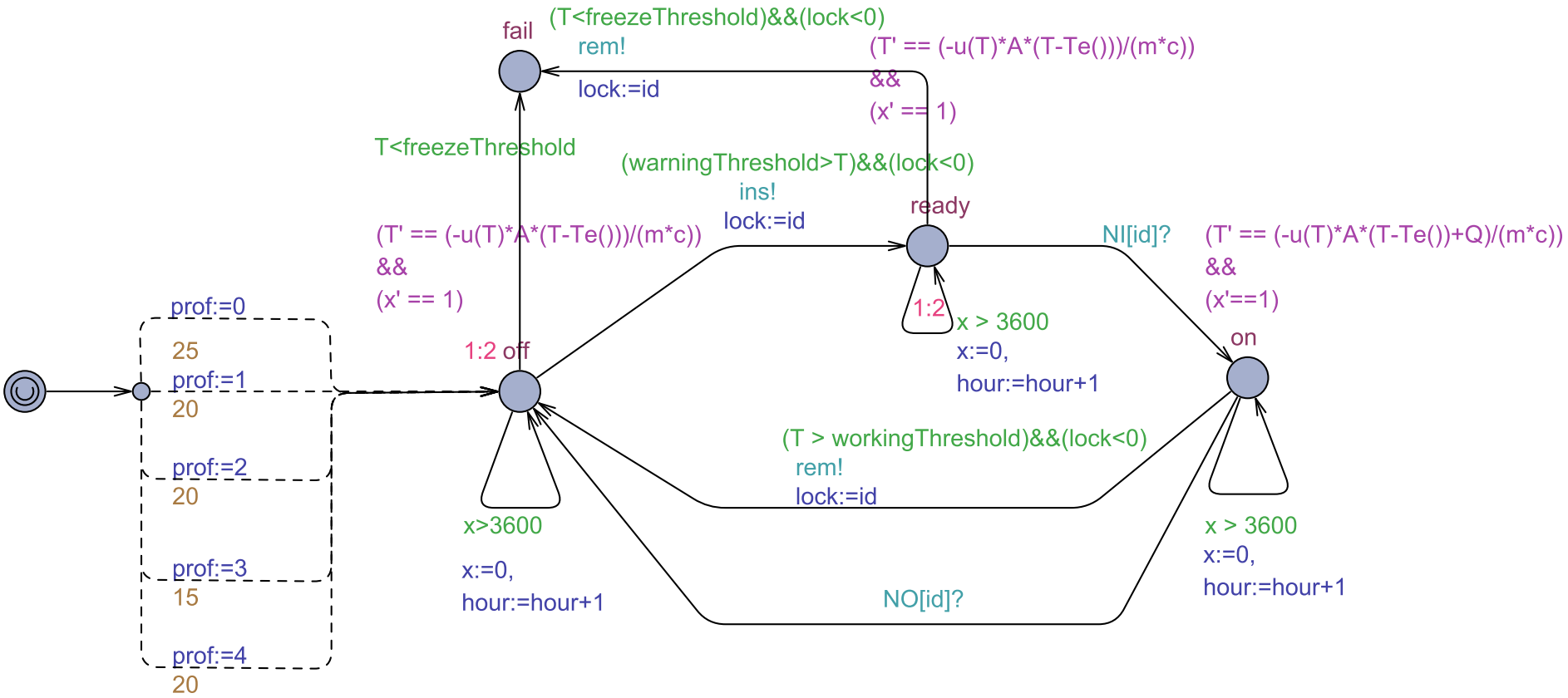
- **Idea**: running a controlled number N of (probabilistically distributed) simulations of a system model to obtain a statistical evaluation p' of some formula φ such that
  - $\Pr(|p' - p| \leq \varepsilon) \geq 1 - \alpha$
  - N only depends on α (sub-linearly) and ε (quadratic)
- **Pros**: avoid full state-space exploration, easy to implement and to parallelise, scalable
- **Cons**: no exact results, difficulties with rare events
- **Uppaal SMC**: frequently used to specify and analyse railway systems, modelled as SHA, implementing SMC
  - graphical interactive simulations, model checking

Basile, ter Beek, Ferrari & Legay, *Modelling and Analysing ERTMS L3 Moving Block Railway Signalling with Simulink and UPPAAL SMC* @ FMICS 2019

Basile, ter Beek & Ciancia, *Statistical Model Checking of a Moving Block Railway Signalling Scenario with Uppaal SMC* @ ISoLA 2018

RailRoadSwitchHeater SHA

- Coordinator $\mathtt{K}$ to manage NH_{max} and priorities
- Composed system: $\mathtt{N} = (\bigotimes_{id \in 1,\ldots,n} \mathtt{H_{id}}) \otimes \mathtt{K}$
- Array of channels for one-to-one communications



$$mc\frac{\partial T}{\partial t} = -uA(T - T_{env}) + \dot{Q}$$

Quantitative analysis

Measures of interest:

- **Energy consumption**: time (in hours) a generic heater is activated in a specific time interval; by multiplying such measurement for the power consumed (kW/h), we can derive the energy consumption
  - Möbius: sum of time spent in specific markings (reward structure)
  - Uppaal: $\mathtt{E}[<= 24; 10000]\ (\mathtt{max} : \sum_{\mathtt{i}:\mathtt{id_t}} \mathtt{H_i.energy})$

- **Reliability**: probability that a generic switch fails (becomes frozen), computed as the probability that no failure occurs in the time interval
  - Möbius: presence of one token at the end of the experiment in the place encoding a failure (reward structure)
  - Uppaal: $\mathbb{P}(\Diamond_{\mathtt{h} \leq 24} \exists(\mathtt{i} : \mathtt{id_t})(\mathtt{H_i.fail}))$

See papers for results (aligned, find parameter setup as best compromise for tradeoff between energy consumption and reliability)

Comparison experiment

We only evaluate primitive support of features, usability and expressiveness

Modelling Features: the composition of, and interactions between, different models (i.e., *heterogeneous formalisms, replicated models, dynamic process instantiation, communication primitives*) and the ability towards modelling hybrid and stochastic systems (i.e., *delay distributions, hybrid variables*)

Properties Specification: the definition of *measures of interest* and the ability to verify properties of the defined models (i.e., *property verification*)

Experiments and Presentation of Results: the setup and execution of experiments, as well as data collection and plotting the results (i.e., *experiments parameter setup*)

Modelling Features

- Replicated models, heterogeneous formalisms, dynamic process instantiation:
  - Möbius: anonymous replicas, hybrid formalism (e.g., SAN, fault trees, PEPA)
  - Uppaal SMC: distinguished replicas, unique formalism (namely SHA), dynamic generation of new processes (fork) available
- Communication primitives, delay distributions, hybrid variables:
  - Möbius: shared memory (places), different stochastic distributions (non-Markovian and Markovian), no primitive support for ODEs
  - Uppaal SMC: channels and shared memory, exponential and uniform distributions, primitive support for ODEs (hybrid systems)

Properties Specification

- Measures of interest, property verification:
  - Möbius: Markov reward models define data to be collected using C++, reward collected based either on the firing of an activity or on marking, collected either at a specific instant, over an (average) interval of time, or upon reaching a steady state
  - Uppaal SMC: a weighted extension of Metric Interval Temporal Logic (i.e., precise semantics), statistical evaluation (e.g., hypothesis testing, probability estimation, probability comparison), supports quantifiers on replicas (cf. measures of interest)

Experiments and Presentation of Results

- Experiments parameter setup:
  - Möbius: experiments organised in batches (called studies) executed in series or in parallel, stores tabular data of experiments' outcome (requiring pre-processing and external tools)
  - Uppaal SMC: single experiments, built-in graphical visualisation of output data and graphical simulation

Comparison between SAN+Möbius and SHA+Uppaal SMC

Features	SAN+Möbius	SHA+Uppaal SMC
Measures of interest	Reward models	MITL formulae
Experiments parameter setup	Batches	Single
Replicated models	Anonymous	Distinguished
Dynamic process instantiation	Not available	Available
Heterogeneous formalisms	Available (SAN, PEPA, etc.)	Not available (SHA)
Communication primitives	Shared places	Channels
Delay distributions	Various distributions	Exponential, uniform
Hybrid variables	No primitive support	ODE solver available
Property verification	Not available	Temporal logics

Discussion on possible future improvements for usability

- Möbius: provide primitive support for
  - non-anonymous replicas and channel communication (already under development by a subset of the authors)
  - graphical visualisation of data, primitive support for ODEs
- Uppaal SMC: provide primitive support for
  - batches of experiments with different parameters
  - other distribution delays (e.g., deterministic time)

Conclusion

- Modelled and analysed an energy saving cyber-physical system in the railway domain stemming from industry, using Möbius with SAN and Uppaal SMC with SHA
- Based on this experience: a comparison of the modelling and analysis capabilities offered by the two frameworks, highlighting some specific features concerning modelling, usability and expressiveness
- Identified a few pointers for future improvements for both frameworks (some already under development)

Thanks for your attention!