A framework for simulation and symbolic state space analysis of non-Markovian models

Laura Carnevali, Lorenzo Ridi, Enrico Vicario

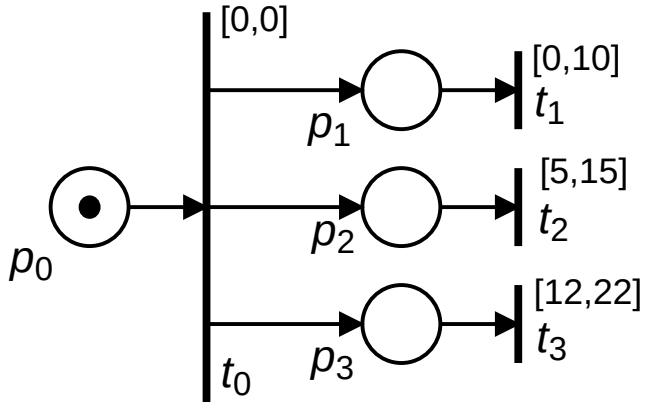
SW Technologies Lab (STLab) - Dip. Sistemi e Informatica (DSI) - Univ. Firenze (UniFI)
{laura.carnevali, lorenzo.ridi, enrico.vicario}@unifi.it
www.stlab.dsi.unifi.it

SafeComp11 - Napoli - September 21, 2011

- this is about a SW framework
  - that supports the construction of tools
    that support the application of formal methods
    that support the development of safe systems
  - encompassing multiple formalisms and solution techniques
    and open to extension and reuse

Time Petri Nets (TPNs)

- concurrent non-deterministic (possibly bounded) timers
  - implicit precedences induced by the timings of concurrent events
  - compare with Timed Automata with location invariants

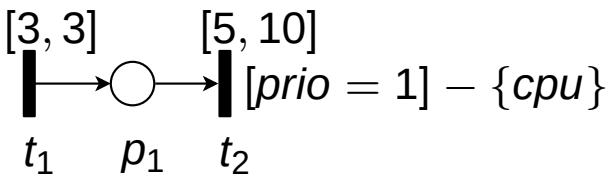


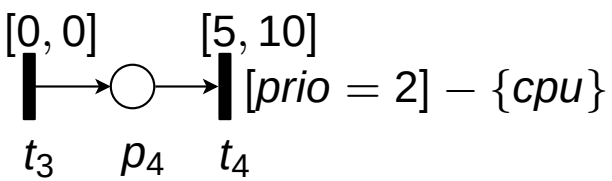
- verification of properties on the set of feasible behaviors
  - event sequencing (e.g., mutual exclusion, deadlock, order)
  - event timing (e.g., min-max Execution Time, deadlines)
- underlying math: Difference Bounds Matrix (DBM) zones

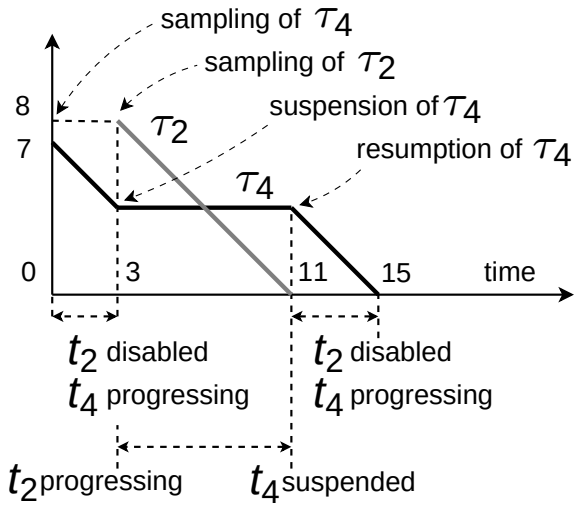
[TSE01] E.Vicario,"Static analysis and dynamic steering of time-dependent systems", IEEE Trans. on SW Eng., 2001.

preemptive Time Petri Nets (pTPNs) 1/2

- represent suspended clocks ($\dot{\tau} = 1$ or $\dot{\tau} = 0$)
  - Preemptive Repeat Different (PRD) and Preemptive Resume (PRs) a.k.a. enabling memory and age policies
    i.e. real time systems running under preemptive scheduling
  - compare with StopWatch Automata and with PNs with hyper-arcs





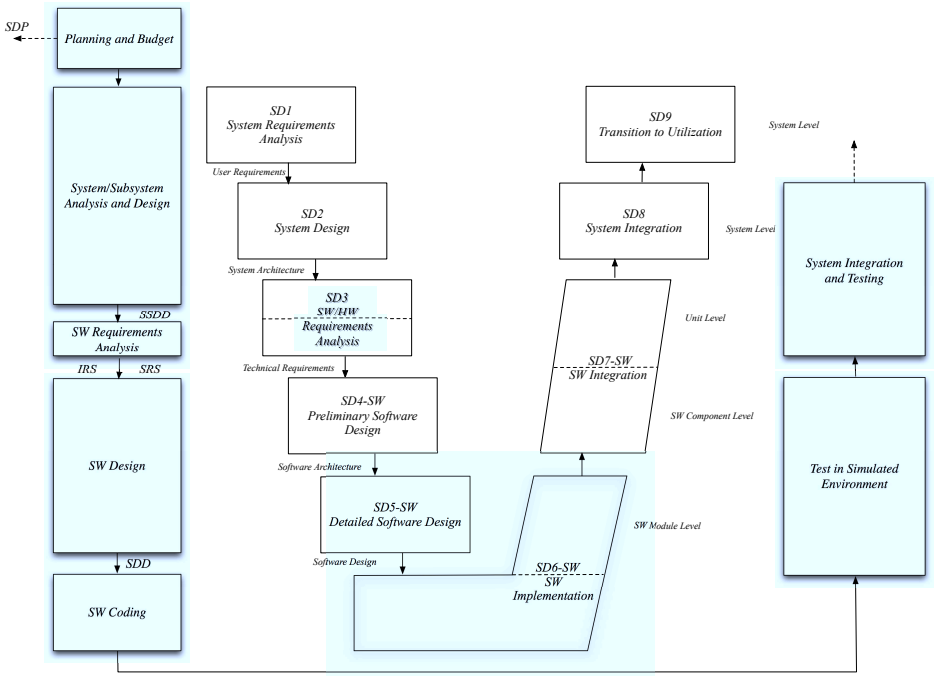


- underlying math: tightest DBM over-approximation with simplex-based clean-up of false behavior

[TSE04] G.Bucci, A.Fedeli, L.Sassoli, and E.Vicario, "Timed state space analysis of real-time preemptive systems", IEEE Trans.on SW Eng. 2004.

preemptive Time Petri Nets (pTPNs) 2/2

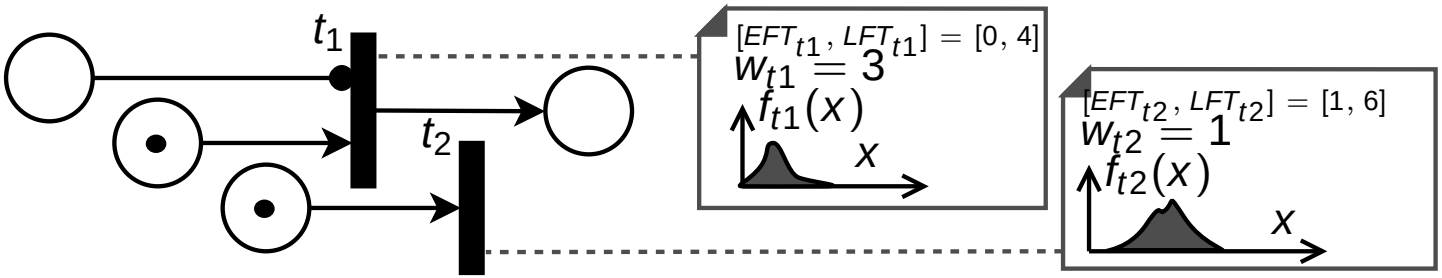
- cast into a V-Model life-cycle
  - preliminary and detailed design + implementation
  - automated code generation, measurement based Execution Time evaluation, formal verification, real time test automation
  - experimented at Selex-Galileo in Florence (MIL-STD-498 and RTCA-DO/178B)



[TSE11] L. Carnevali, L. Ridi, and E. Vicario, "Putting preemptive Time Petri Nets to work in a V-Model SW life cycle", IEEE Trans. on SW Eng., accepted for publication

stochastic Time Petri Nets (sTPNs)

- a class of non-Markovian Stochastic Petri Nets
  - multiple concurrent GEN transitions with possibly bounded support
  - the underlying stochastic process may become Markov Regenerative (MRGP) or Generalized Semi-Markov (GSMP)



- underlying math: stochastic state classes (i.e. symbolic derivation of probability density functions over DBM zones)

[QEST11] A.Horvath, L.Ridi, E.Vicario, "Probabilistic model checking of non-Markovian models with concurrent generally distributed timers", QEST 2011.

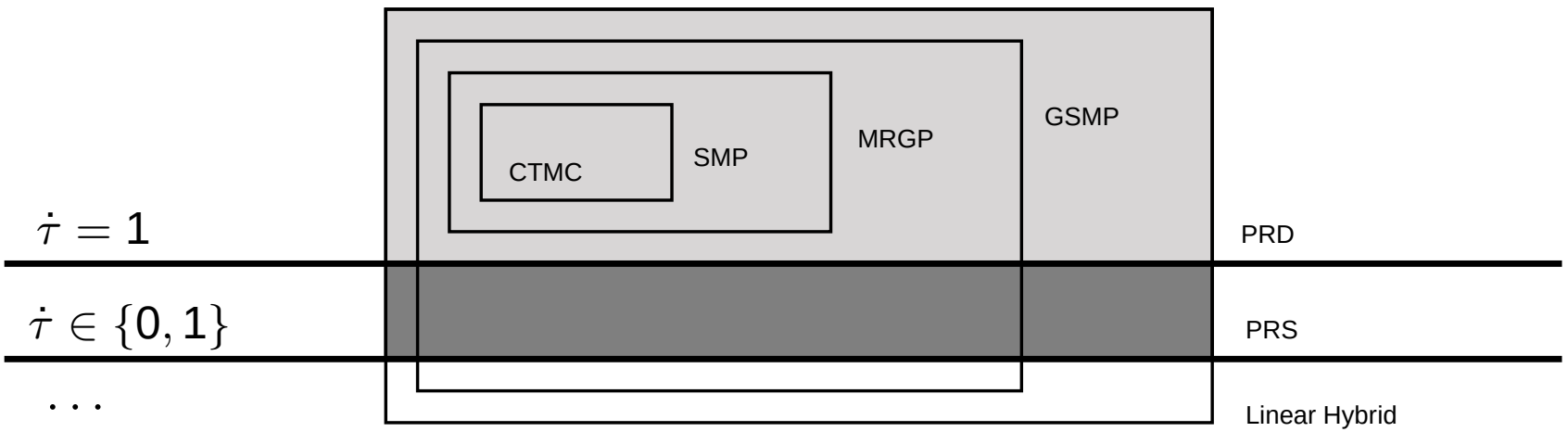
[QEST10] A.Horvath, L.Ridi, E.Vicario, "Transient analysis of generalised semi-Markov processes using transient stochastic state classes", QEST 2010.

[TSE09b] E.Vicario, L.Sassoli, L.Carnevali, "Using Stochastic State Classes in Quantitative Evaluation of Dense-Time Reactive Systems", IEEE Trans.SW Eng.2009.

[TSE09a] L.Carnevali, L.Grassi, E.Vicario, "State-density functions over DBM domains in the analysis of non-Markovian models", IEEE Trans.SW Eng.2009.

stochastic preemptive Time Petri Nets (spTPNs)

- stochastic Time Petri Nets with clock suspension
  - aiming at application in soft-real time systems

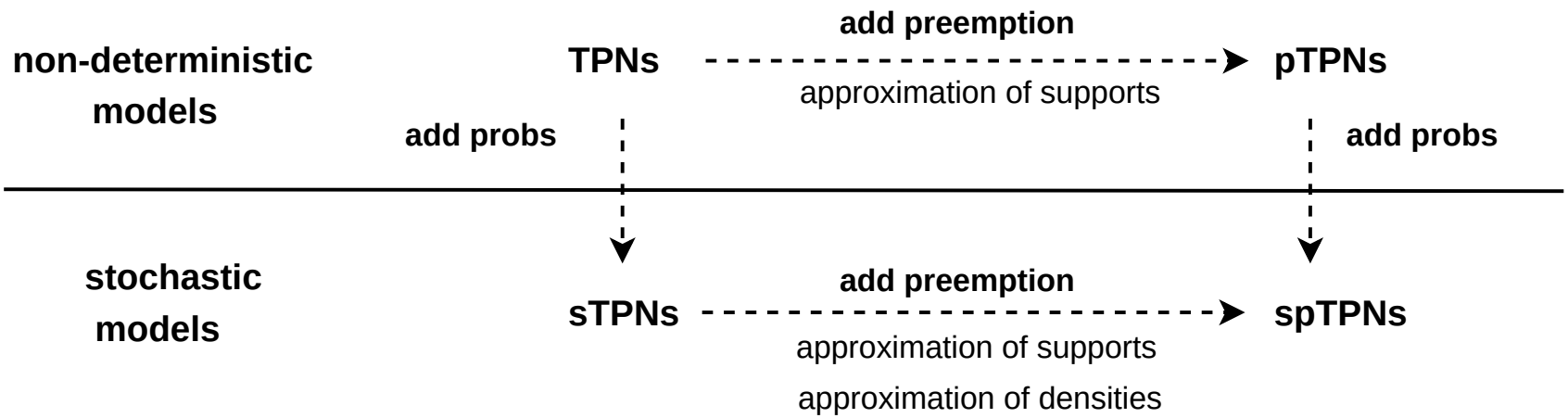


- underlying math: approximation on stochastic state classes

[VT11] L. Carnevali, J. Giuntini, and E. Vicario, "A symbolic approach to quantitative analysis of preemptive real-time systems with non-Markovian temporal parameters", ValueTools 2011.

the overall picture

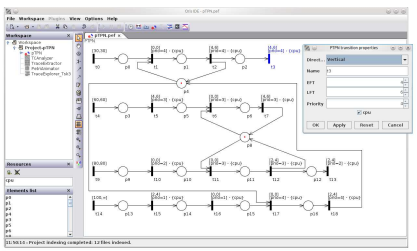
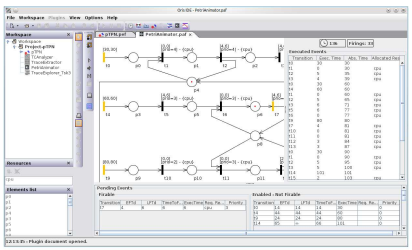
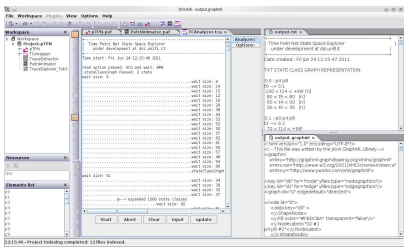
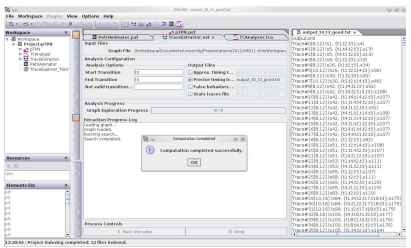
- a number of modeling formalisms and solution techniques
  - non-deterministic vs probabilitic
  - cooperative vs preemptive



- ... and more could come
  - continuous time vs discrete time
  - mixed probabilistic and non-deterministic
  - ...

the Oris Tool

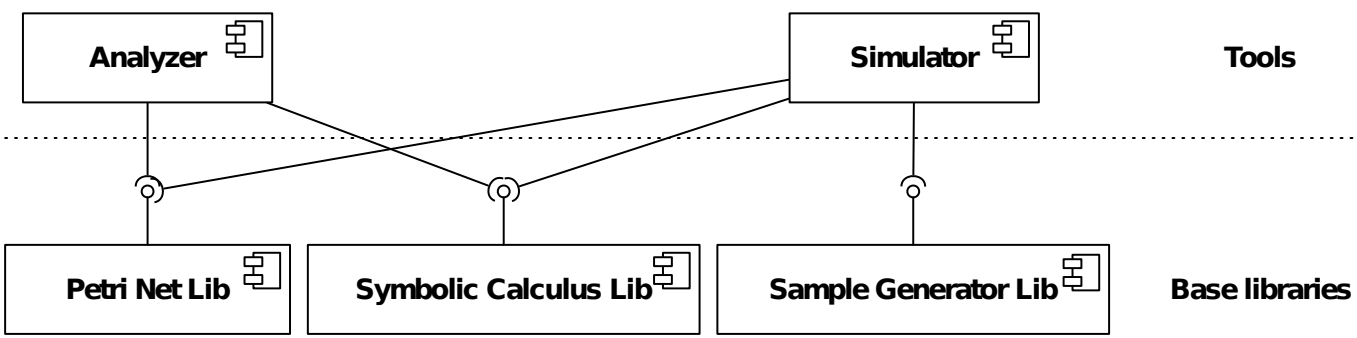
- a tool implementing all the theory developed along the years
  - available at `www.stlab.dsi.unifi.it/oris`
  - manages correctness verification and quantitative evaluation of timed concurrent systems specified as Petri Net models, i.e., TPNs, pTPNs, sTPNs, spTPNs

- ... with growing effort in keeping it all together
  - refactoring/porting to full Java implementation
  - based on a well engineered framework, with hinges enabling adaptation and evolution

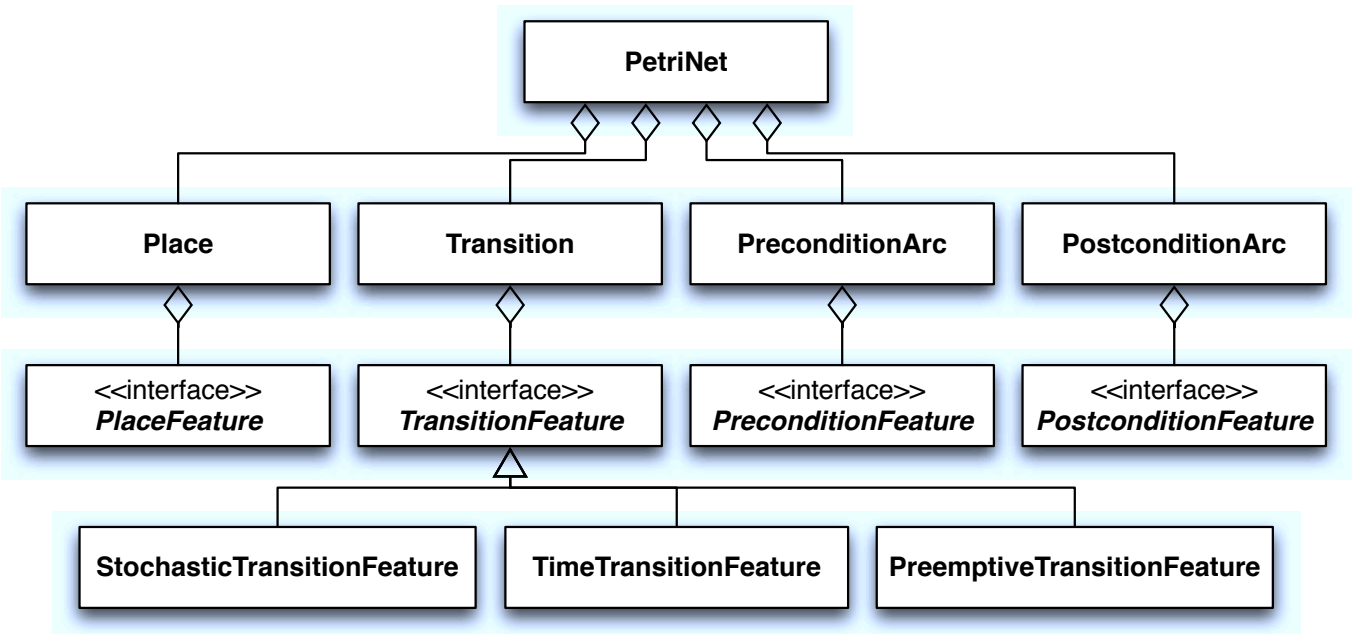
Sirio: functional responsibilities

- base libraries: representation of model syntax
  - Petri Net Library
  - Symbolic Calculus Library
  - Sample Generator Library
  - open to extensions for modeling formalisms
- tools: implementation of model semantics and analysis methods
  - Simulator Tool
  - Analyzer Tool
  - open to extensions for solutions techniques



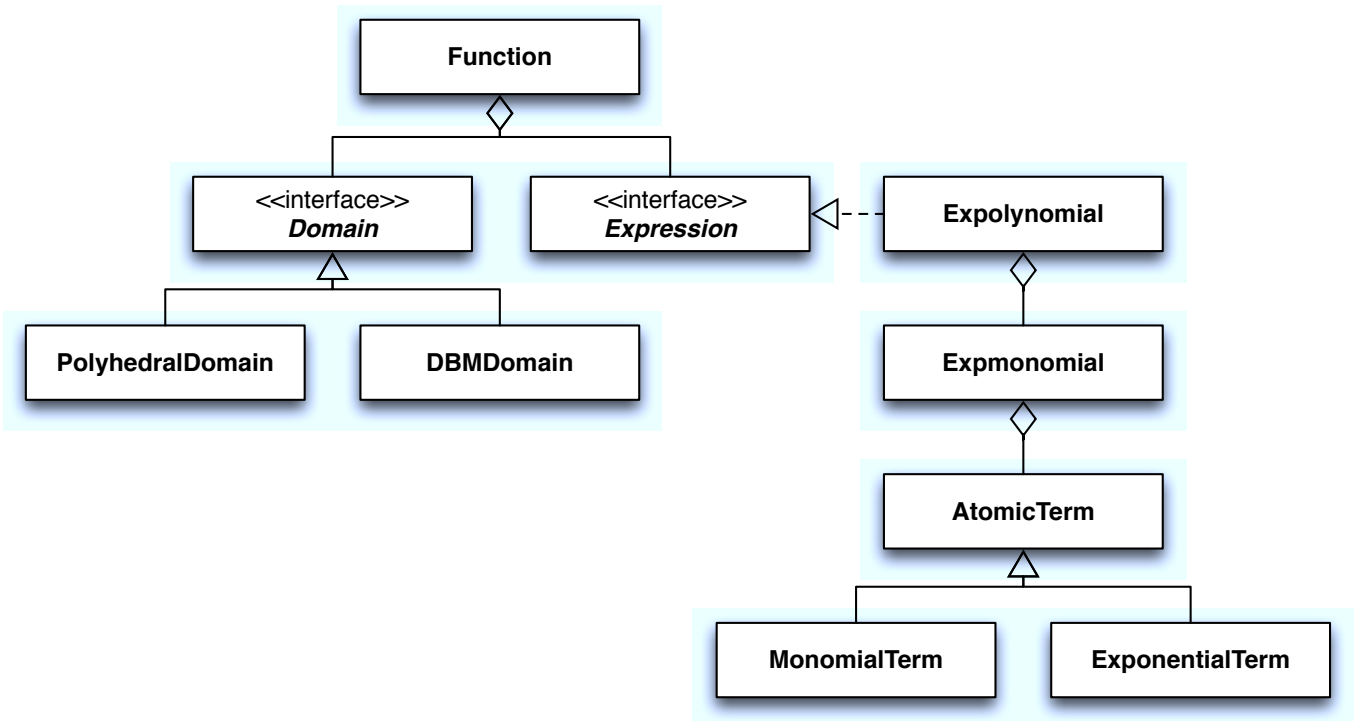
Petri Net Library

- supports representation of TPNs, pTPNs, sTPNs, and spTPNs
- the type of a Petri Net model is *implicitly* determined by the features associated with its structural components
- the implementation of new Petri Net models amounts to the implementation of new features for its components



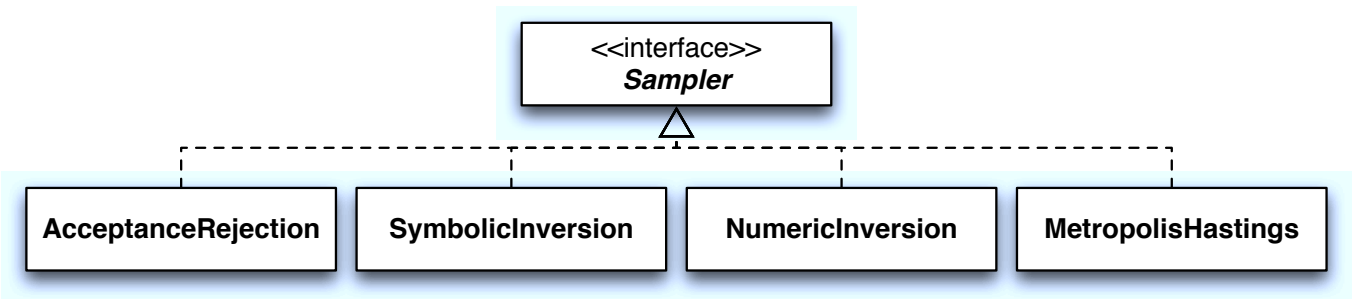
Symbolic Calculus Library

- supports definition and manipulation of expolynomial functions over polyhedral and Difference Bounds Matrix (DBM) domains



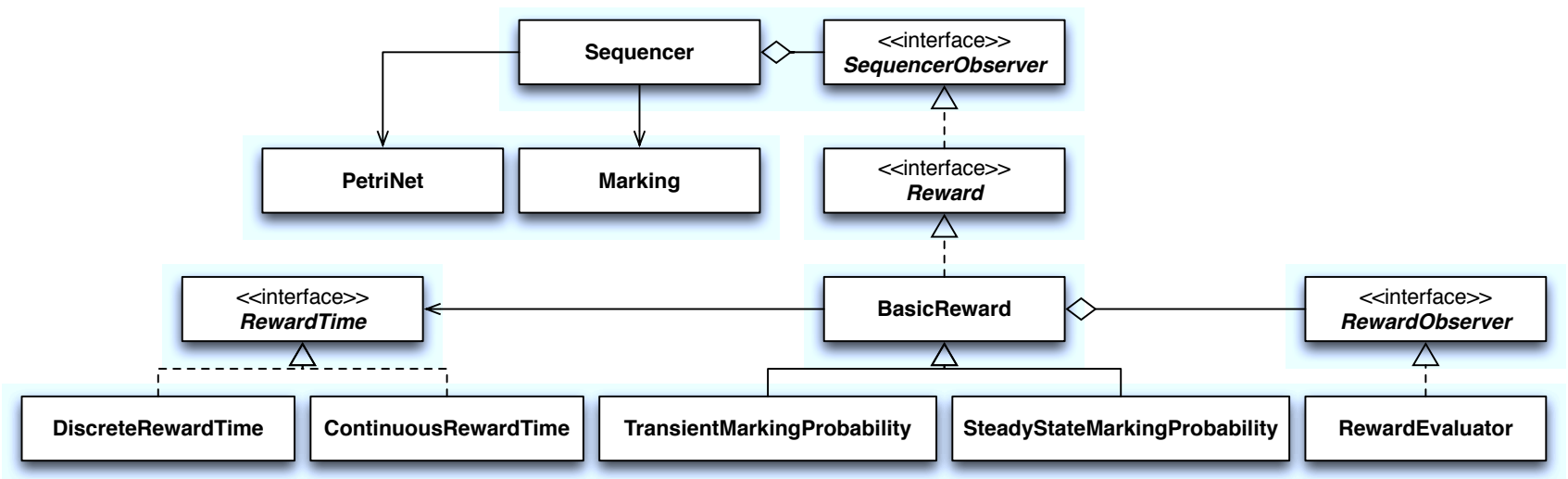
Sample Generator Library

- implements generation of samples from expolynomial distributions
  using different methods depending on the form of the distribution, e.g., symbolic/numeric inversion, Metropolis-Hastings algorithm



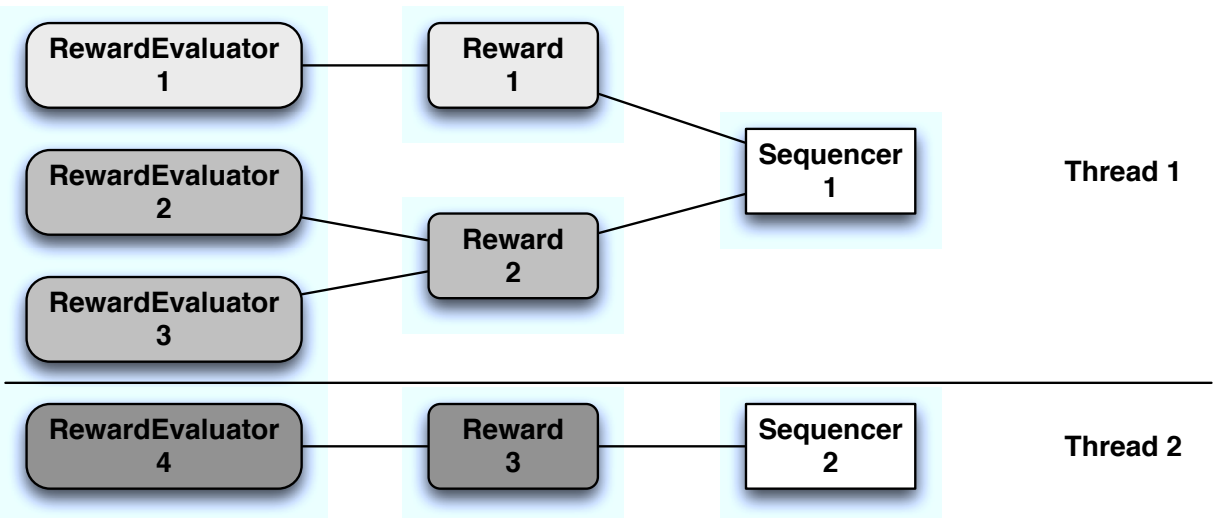
Simulator Tool

- implements the semantics of TPNs, pTPNs, sTPN, and spTPNs
- supports evaluation of *transient* and *steady state rewards*, both in *Discrete Time* (DT) and in *Continuous Time* (CT)
- uses composition instead of inheritance to decouple the reward interface from its implementation in DT/CT
- the definition of new reward measures amounts to the definition of a class that extends `BasicReward` and (possibly) the addition of methods to `RewardTime`



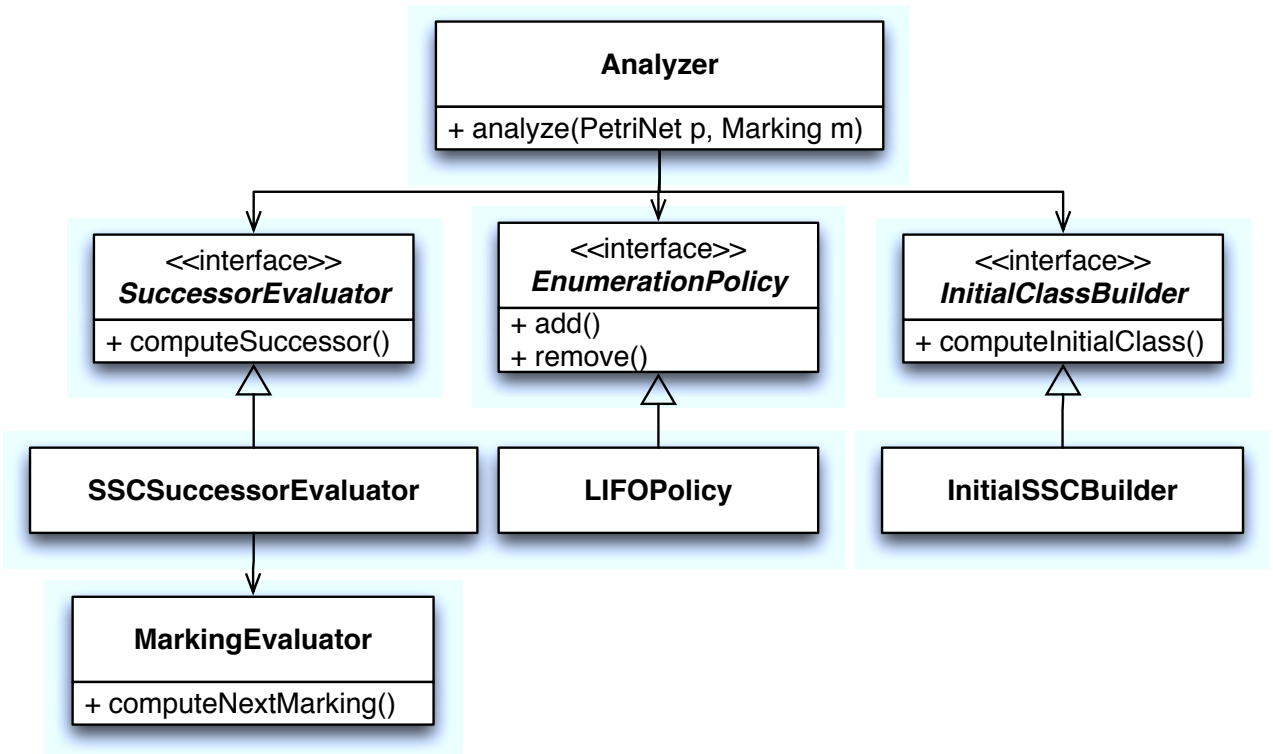
Simulator Tool

- implements the semantics of TPNs, pTPNs, sTPN, and spTPNs
- supports evaluation of *transient* and *steady state rewards*, both in *Discrete Time* (DT) and in *Continuous Time* (CT)
- uses composition instead of inheritance to decouple the reward interface from its implementation in DT/CT
- the definition of new reward measures amounts to the definition of a class that extends `BasicReward` and (possibly) the addition of methods to `RewardTime`



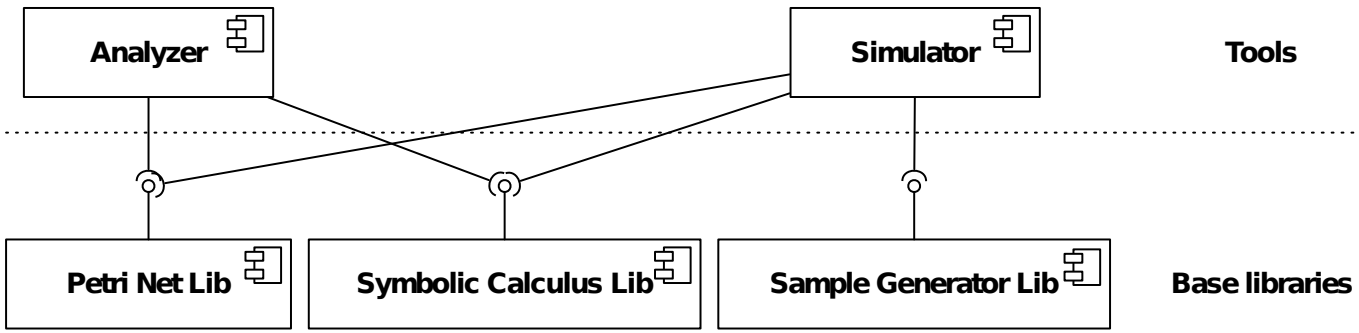
Analyzer Tool

- implements the theory of stochastic state classes
- uses composition instead of inheritance to customize the behavior of a general enumeration algorithm
- the implementation of new techniques of analysis amounts to the definition of delegate components for the `Analyzer` class



conclusions

- a Java framework for verification and evaluation of timed concurrent systems specified as Petri Net models



- implementing verification and evaluation of non-deterministic and Stochastic (possibly non-Markovian) Petri Nets
  - exploited in the Oris tool
- open to extension to other formalisms
  - and available for integration in any other tools
- `www.stlab.dsi.unifi.it/oris`