The ORIS Tool: App, Library and Toolkit for Quantitative Evaluation of Non-Markovian Systems

Laura Carnevali [1] Marco Paolieri [2] Enrico Vicario [1]

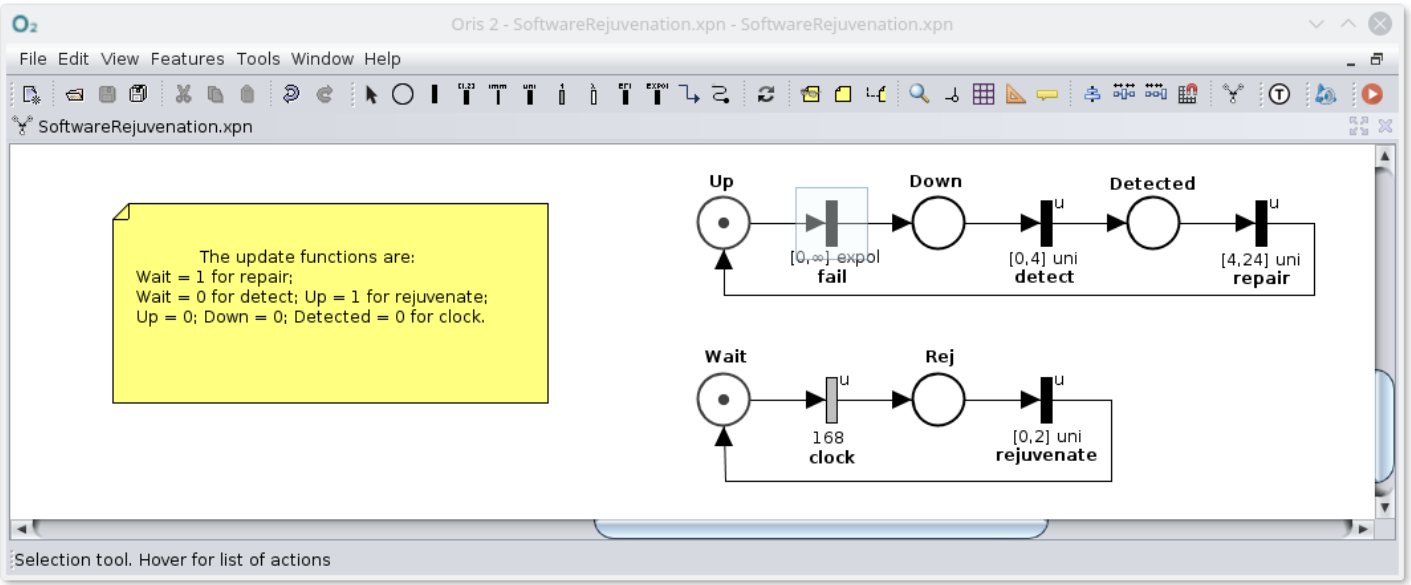
[1]Department of Information Engineering, University of Florence, Italy
[2]Department of Computer Science, University of Southern California, USA

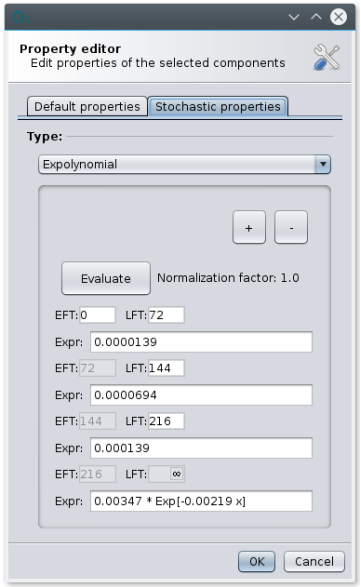
TOSME 2021: Workshop on Tools for Stochastic Modelling and Evaluation
November 12, 2021

Extended contents about ORIS are reported in:
The ORIS Tool: Quantitative Evaluation of Non-Markovian Systems
by Marco Paolieri, Marco Biagi, Laura Carnevali, Enrico Vicario
IEEE Trans. on Software Engineering, vol. 47, no. 6, pp. 1211–1225, June 2021
DOI: 10.1109/TSE.2019.2917202

ORIS as a GUI Application for Models of Stochastic Systems

- GUI application for **quantitative evaluation** of **stochastic models**
  - Model specification using Stochastic Time Petri Nets (STPNs)
  - Model validation using interactive simulation (token game)
  - Transient/steady-state model analysis using different analysis engines

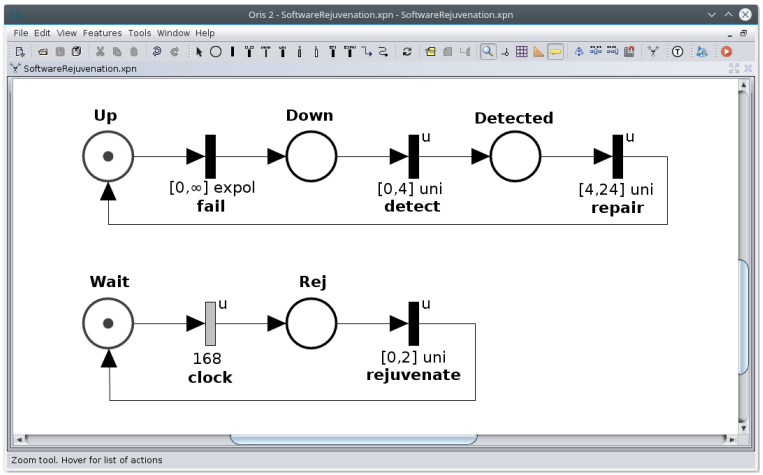


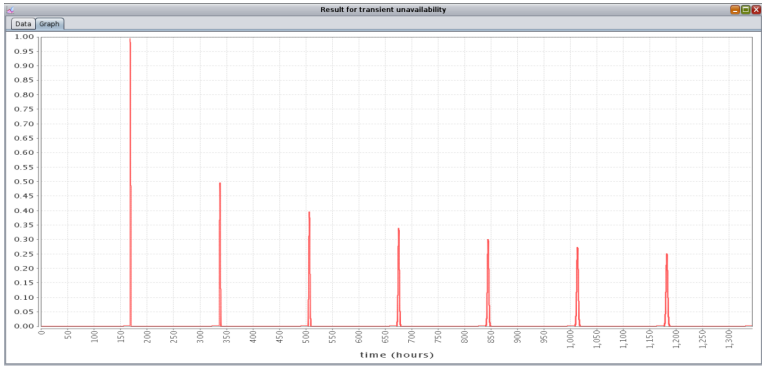


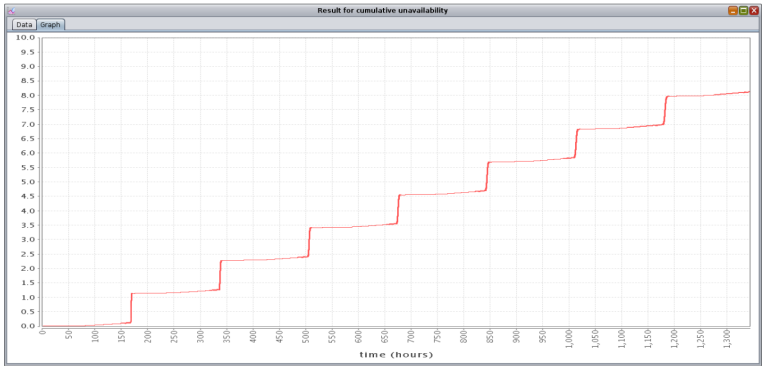
Evaluation of Quantitative Properties in ORIS

- **Transient unavailability**
  - Transient probability that the system is not working at time t
  - Instantaneous transient reward `Down>0||Detected>0||Rej>0` (regenerative transient analysis, time limit 1344 h, step size 0.005 h)

- **Cumulative unavailability**
  - Expected outage time within the time interval $[0, t]$
  - Cumulative transient reward `Down>0||Detected>0||Rej>0` (regenerative transient analysis, time limit 1344 h, step size 0.005 h)







Analysis Engines in ORIS

Different classes of **stochastic processes** supported:

- Only exponential timers: **Continuous-Time Markov Chain** (CTMC)
- General timers reset after each firing: **Semi-Markov Process** (SMP)
- General timers reset after *n* firings: **Markov Regenerative Process** (MRP)
- No restrictions: **Generalized Semi-Markov Process** (GSMP)

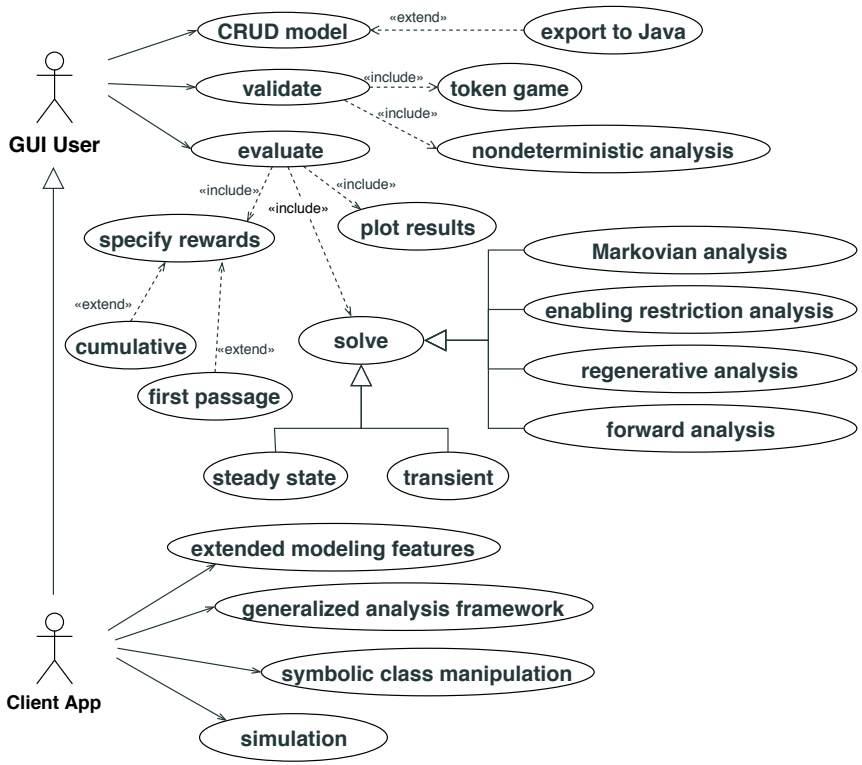
Papers on the **solution techniques** implemented in ORIS:

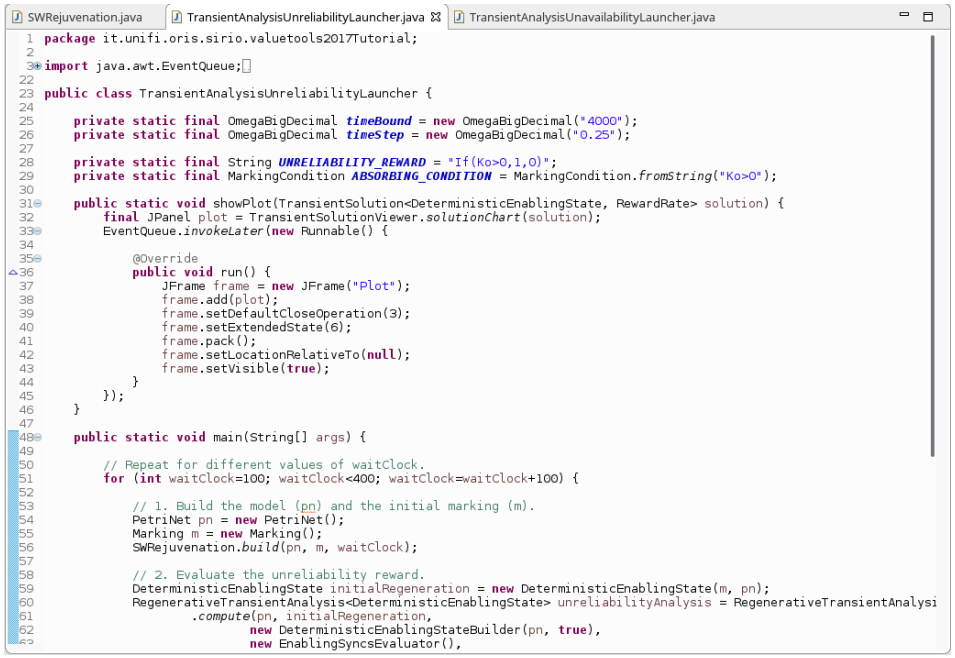
- *Probabilistic model checking of regenerative concurrent systems*
  M. Paolieri, A. Horváth, E. Vicario, IEEE Trans. on Software Engineering, 2016
- *Transient analysis of non-Markovian models using stochastic state classes*
  A. Horváth, M. Paolieri, L. Ridi, E. Vicario, Performance Evaluation, 2012
- *Using Stochastic State Classes in Quantitative Evaluation of Dense-Time Reactive Systems*
  E. Vicario, L. Sassoli, L. Carnevali, IEEE Trans. on Software Engineering, 2009
- *State-Density Functions over DBM Domains in the Analysis of Non-Markovian Models*
  L. Carnevali, L. Grassi, E. Vicario, IEEE Trans. on Software Engineering, 2009
- *Static Analysis and Dynamic Steering of Time-Dependent Systems Using Time Petri Nets*
  E. Vicario, IEEE Trans. on Software Engineering, 2001

ORIS as a Java Library

ORIS provides a Java library (SIRIO) for modeling and analysis of STPNs

- To carry out **extensive performance and reliability studies**
- To implement **models@runtime**
- To implement new **model features** and new **analysis methods**

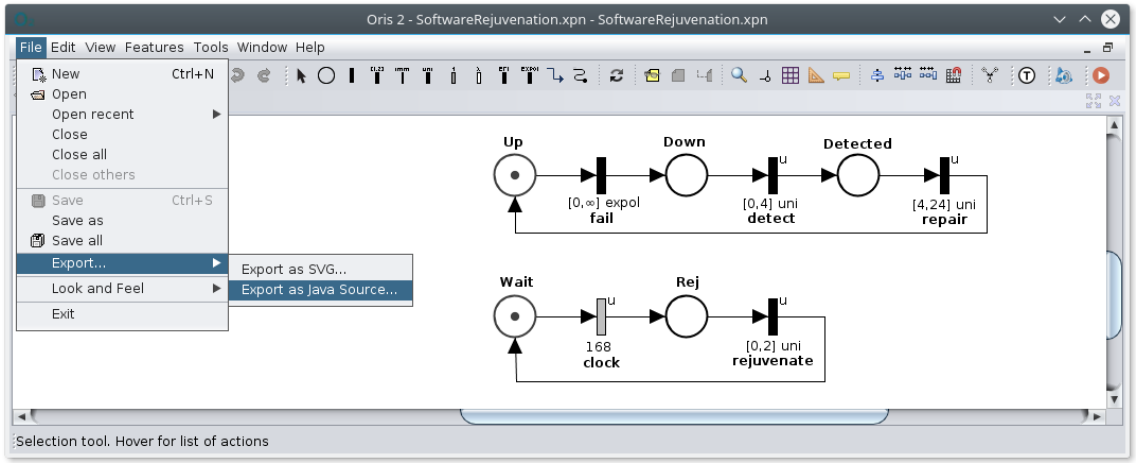


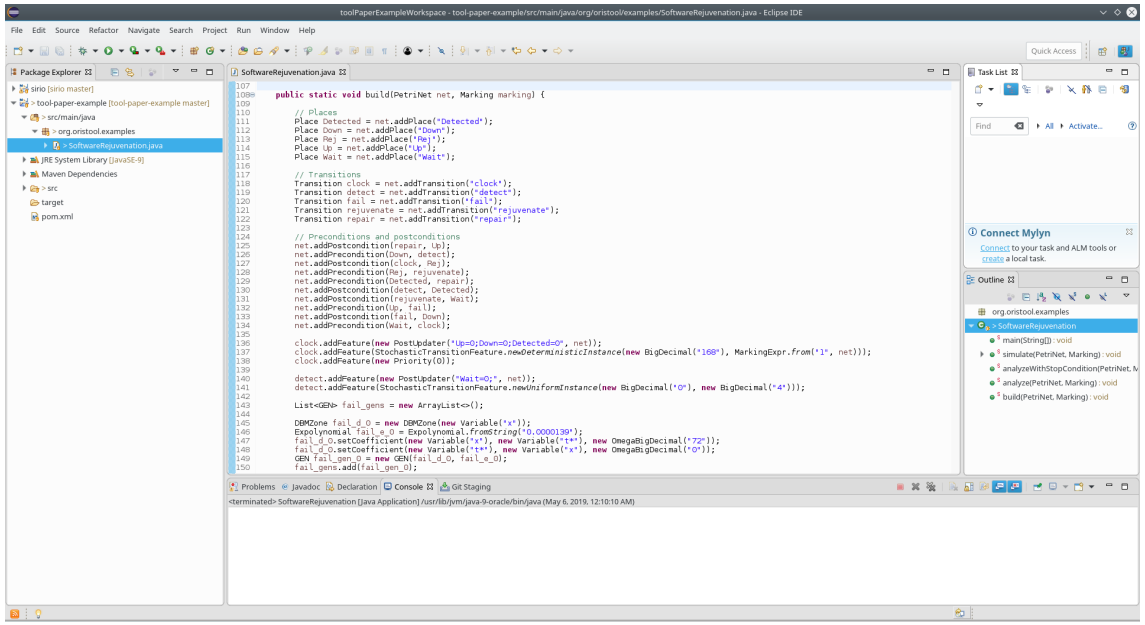



```java
package it.unifi.oris.sirio.valuetools2017Tutorial;

import java.awt.EventQueue;

public class TransientAnalysisUnreliabilityLauncher {

    private static final OmegaBigDecimal timeBound = new OmegaBigDecimal("4000");
    private static final OmegaBigDecimal timeStep = new OmegaBigDecimal("0.25");

    private static final String UNRELIABILITY_REWARD = "If(Ko>0,1,0)";
    private static final MarkingCondition ABSORBING_CONDITION = MarkingCondition.fromString("Ko>0");

    public static void showPlot(TransientSolution<DeterministicEnablingState, RewardRate> solution) {
        final JPanel plot = TransientSolutionViewer.solutionChart(solution);
        EventQueue.invokeLater(new Runnable() {

            @Override
            public void run() {
                JFrame frame = new JFrame("Plot");
                frame.add(plot);
                frame.setDefaultCloseOperation(3);
                frame.setExtendedState(6);
                frame.pack();
                frame.setLocationRelativeTo(null);
                frame.setVisible(true);
            }
        });
    }

    public static void main(String[] args) {

        // Repeat for different values of waitClock.
        for (int waitClock=100; waitClock<400; waitClock=waitClock+100) {

            // 1. Build the model (pn) and the initial marking (m).
            PetriNet pn = new PetriNet();
            Marking m = new Marking();
            SWRejuvenation.build(pn, m, waitClock);

            // 2. Evaluate the unreliability reward.
            DeterministicEnablingState initialRegeneration = new DeterministicEnablingState(m, pn);
            RegenerativeTransientAnalysis<DeterministicEnablingState> unreliabilityAnalysis = RegenerativeTransientAnalysi
                    .compute(pn, initialRegeneration,
                            new DeterministicEnablingStateBuilder(pn, true),
                            new EnablingSyncsEvaluator(),
```

ORIS as a Toolkit

- Define/validate model with app and **export it as Java code** that uses SIRIO

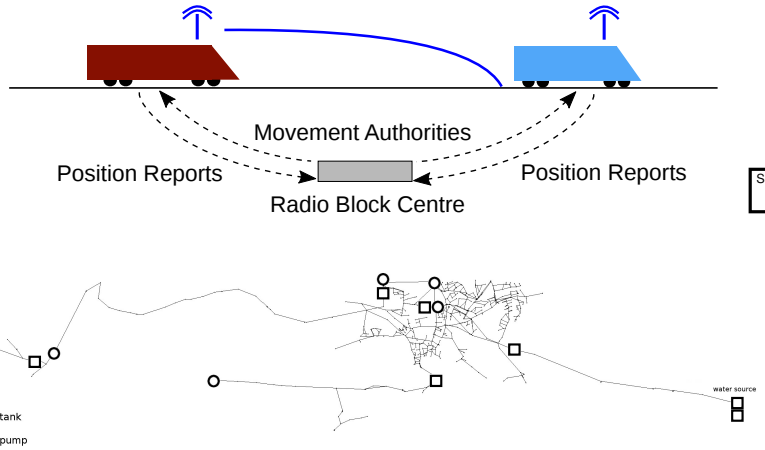


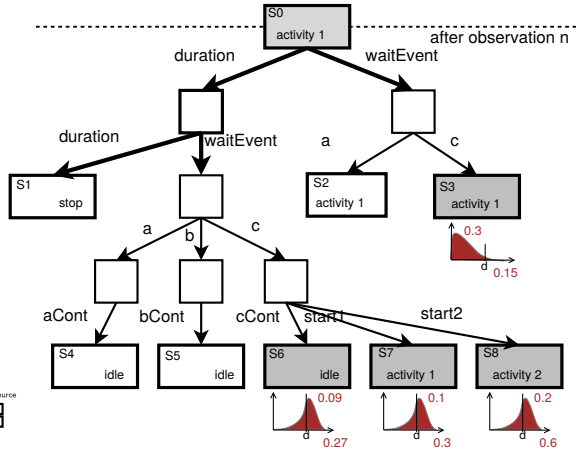


Applications using ORIS

Papers on **applications** of stochastic models analyzed with ORIS:

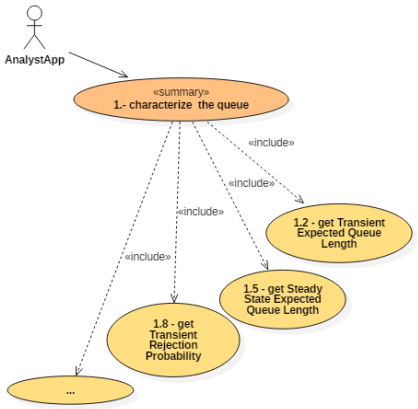
- *Performability evaluation of the ERTMS/ETCS - Level 3*
  M. Biagi, L. Carnevali, M. Paolieri, E. Vicario, Transportation Research: Part C, 2017
- *A Quantitative Approach to Input Generation in Real-Time Testing of Stochastic Systems*
  L. Carnevali, L. Ridi, E. Vicario, IEEE Trans. on Software Engineering, 2013
- *Performability Evaluation of Water Distribution Systems During Maintenance Procedures*
  L. Carnevali, F. Tarani, E. Vicario, IEEE Trans. on Sys., Man, and Cybern. Systems, 2020
- *Model-based quantitative evaluation of repair procedures in gas distribution networks*
  M. Biagi, L. Carnevali, F. Tarani, E. Vicario, ACM Trans. on Cyber-Physical Systems, 2018
- *A continuous-time model-based approach for activity recognition in pervasive environments*
  M. Biagi, L. Carnevali, M. Paolieri, F. Patara, E. Vicario, IEEE Trans. Human-Machine Sys., 2019

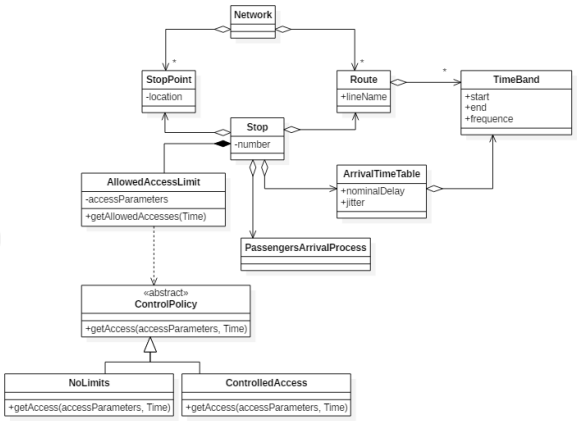


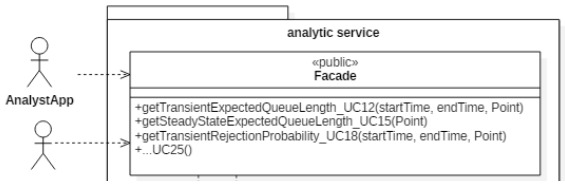


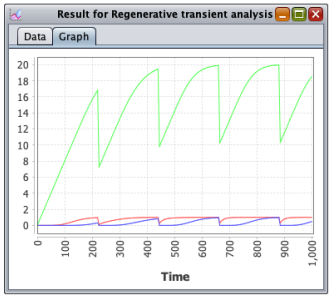
Using ORIS to support Model Driven Engineering: from Domain Metamodels to Runtime Analysis

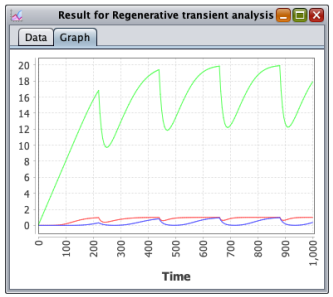
- Manual **design of STPN model** and **export as Java** with ORIS app
- Object-oriented **domain model** visited by **model builder**
- Facade to **evaluate metrics** for use cases of **system analysts** and **users**

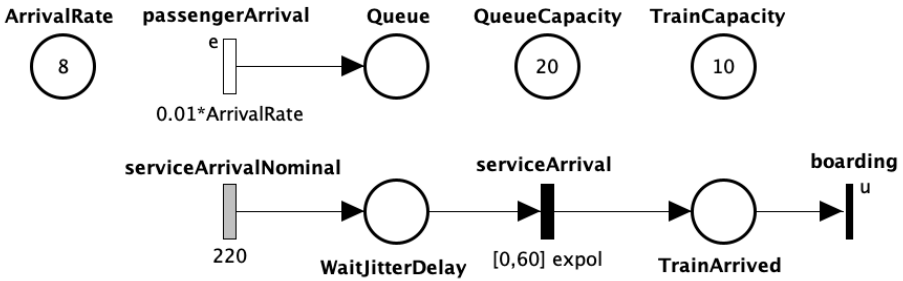












Obtaining ORIS

- Freely available (source code released under AGPL license)
- Downloads, tutorials, API documentation at `www.oris-tool.org`
- Java library examples at `github.com/oris-tool/sirio-examples`