

REMES Tool-chain: A Set of Integrated Tools for Behavioural Modelling and Analysis of Embedded Systems

Dinko Ivanov¹

Marin Orlić²

Cristina Seceleanu³

Aneta Vulgarakis³

¹*University "St.Kliment Ohridski", Bulgaria*

²*University of Zagreb, Croatia*

³*Mälardalen University, Sweden*

Introduction

- Embedded systems (ES)
 - Resource constrained
 - Time constrained, must be checked (formal analysis)
- Component-based approach to design of embedded systems
- Goal: support modelling and analysis of ES as early in the design phase as possible
 - Modelling architecture + behaviour
- Predictability – top challenge in ES



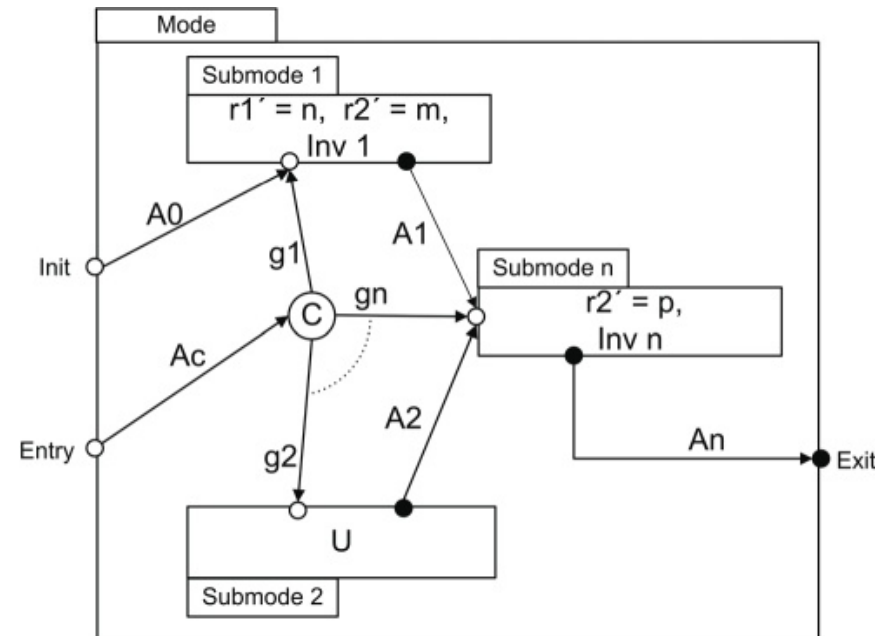
Motivation



- Focus on behaviour modelling
 - Functionality, timing constraints, resource usage
- Enable model input, support model-checking
- Introduce simulation
 - Spot logical errors early in the design phase
 - Perfect the model before performing full formal analysis
 - Test prototype model quickly
- Contribution
 - Tool-chain: editor, simulator, mapping to formal model

REMES behavioural language

- REMES (REsource Model for Embedded System) resource and behaviour model
 - Functional behaviour (discrete state-based)
 - Resource behaviour (discrete, continuous)
 - Timing behaviour (dense time, state based)
 - (Priced) Timed automata (TA / PTA) for analysis
 - Separation between input and output
 - Resources as primitives
 - Explicit types

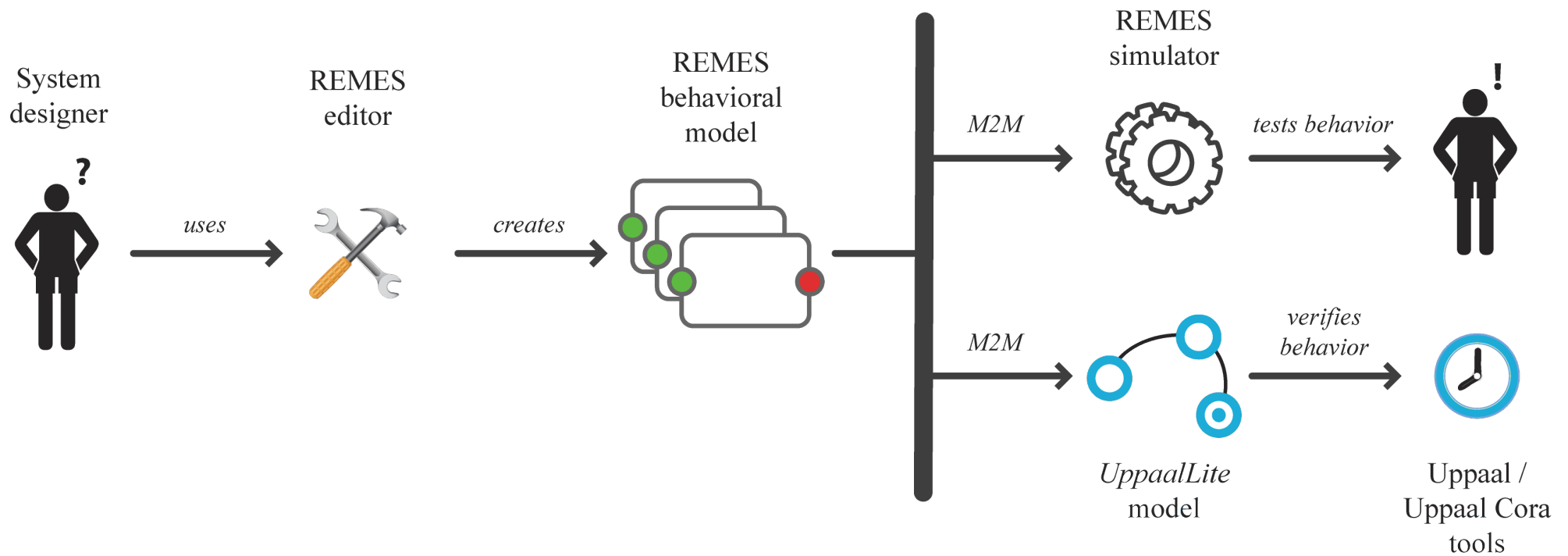


The toolchain

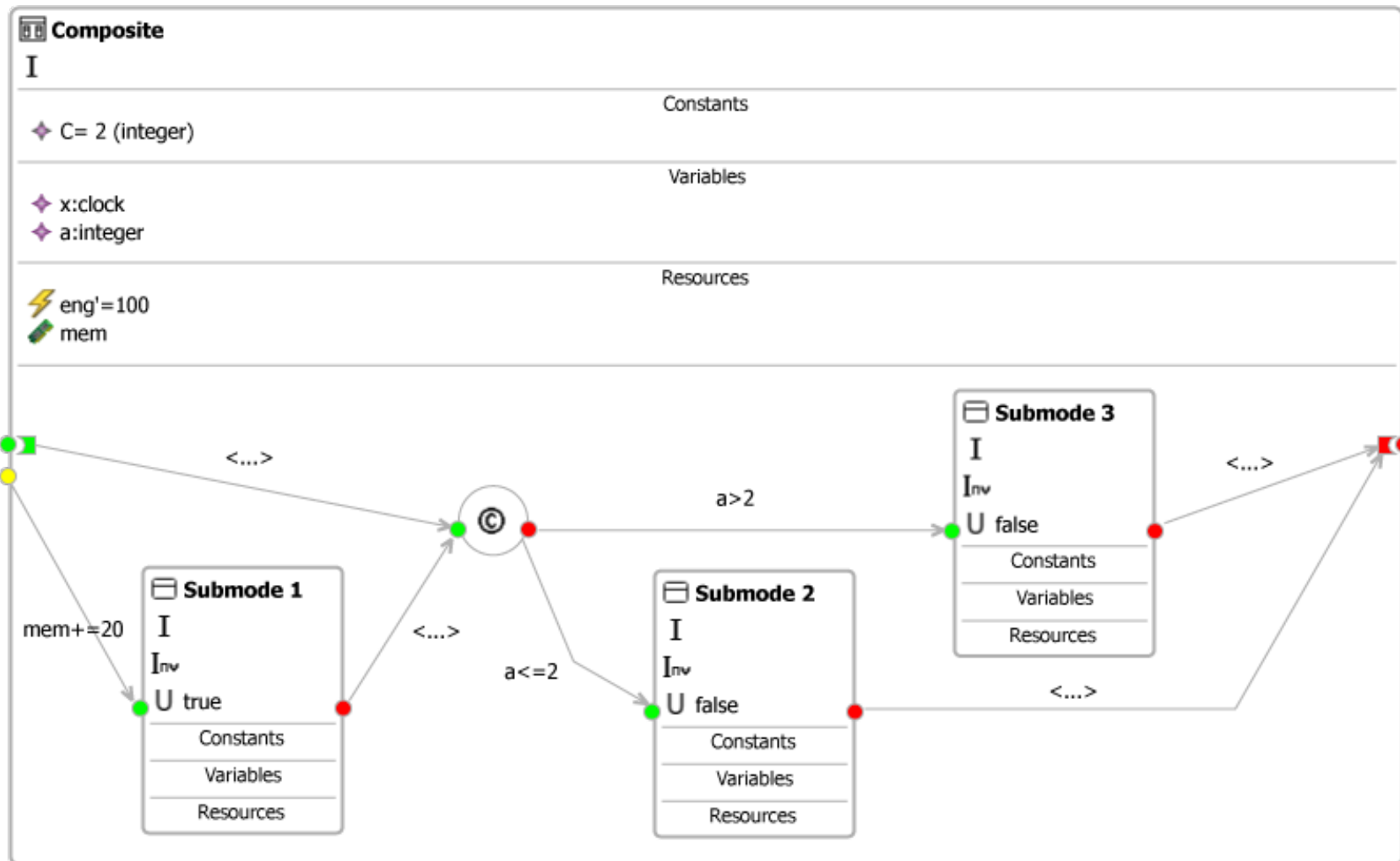
- Built on Eclipse Platform
 - Integrates with PRIDE – IDE for component-based ES development
- Graphical REMES model editor
- Simulator
- Transformation to analytical model



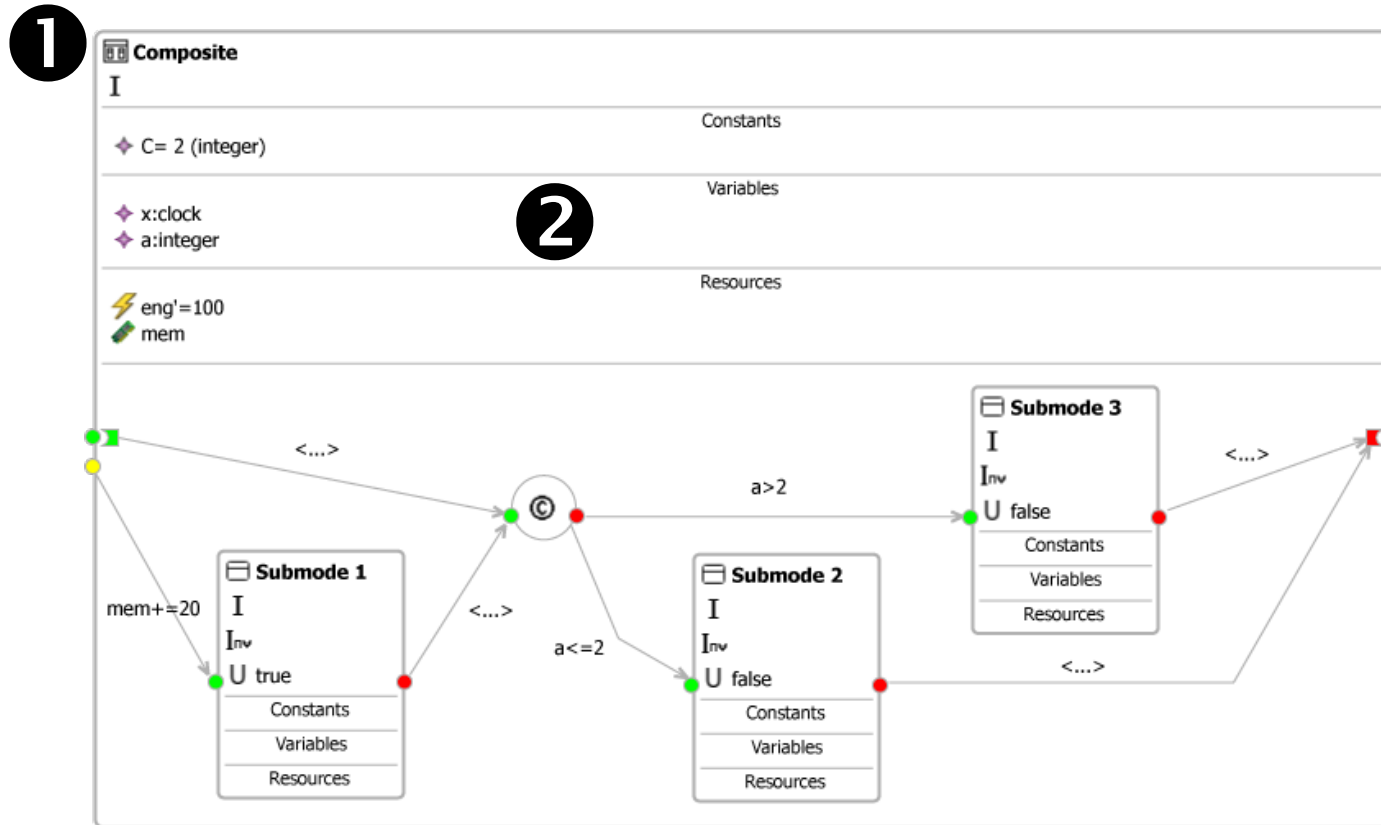
Workflow



I. REMES language editor

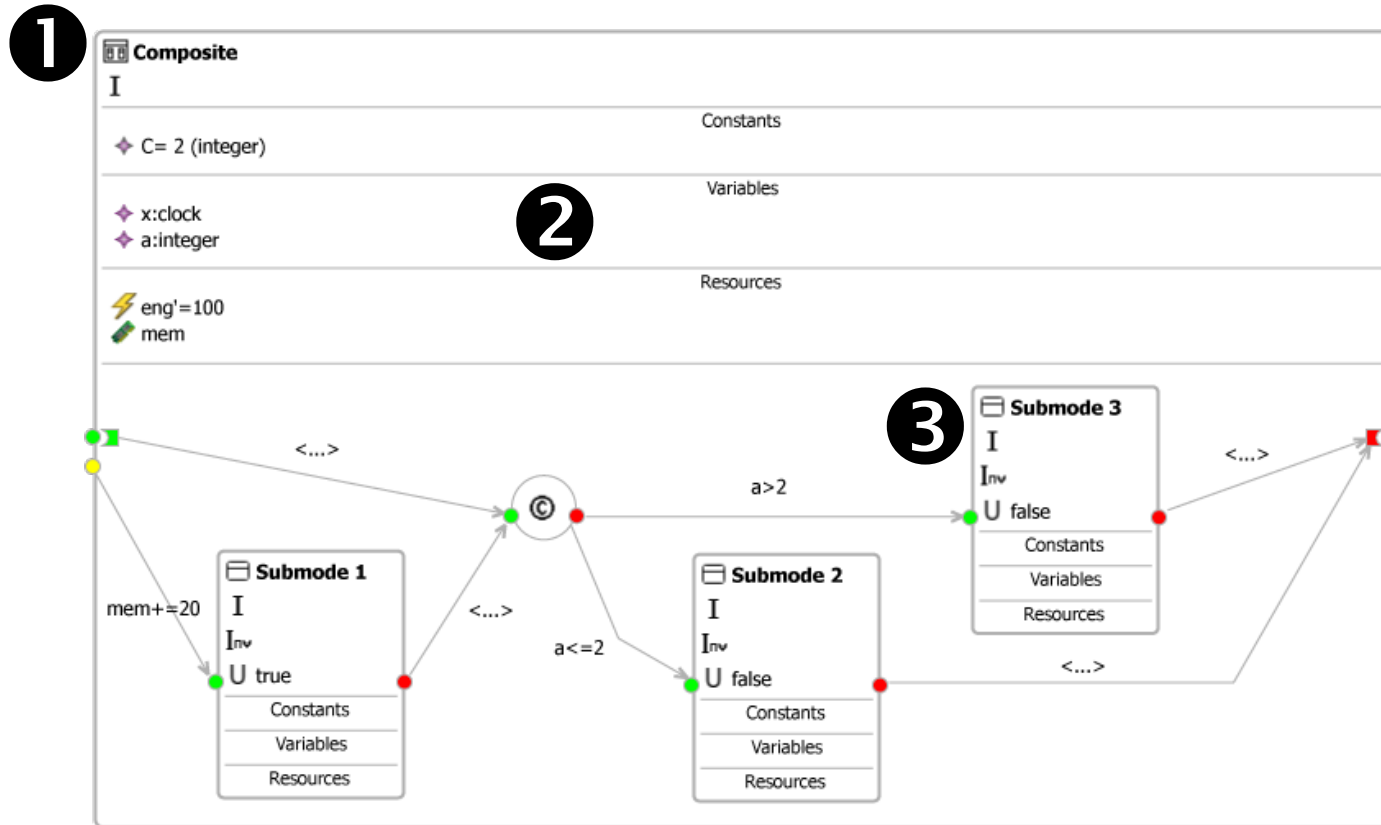


REMES language elements



- Composite mode **1**
- Compartments **2** for variables, resources, constants

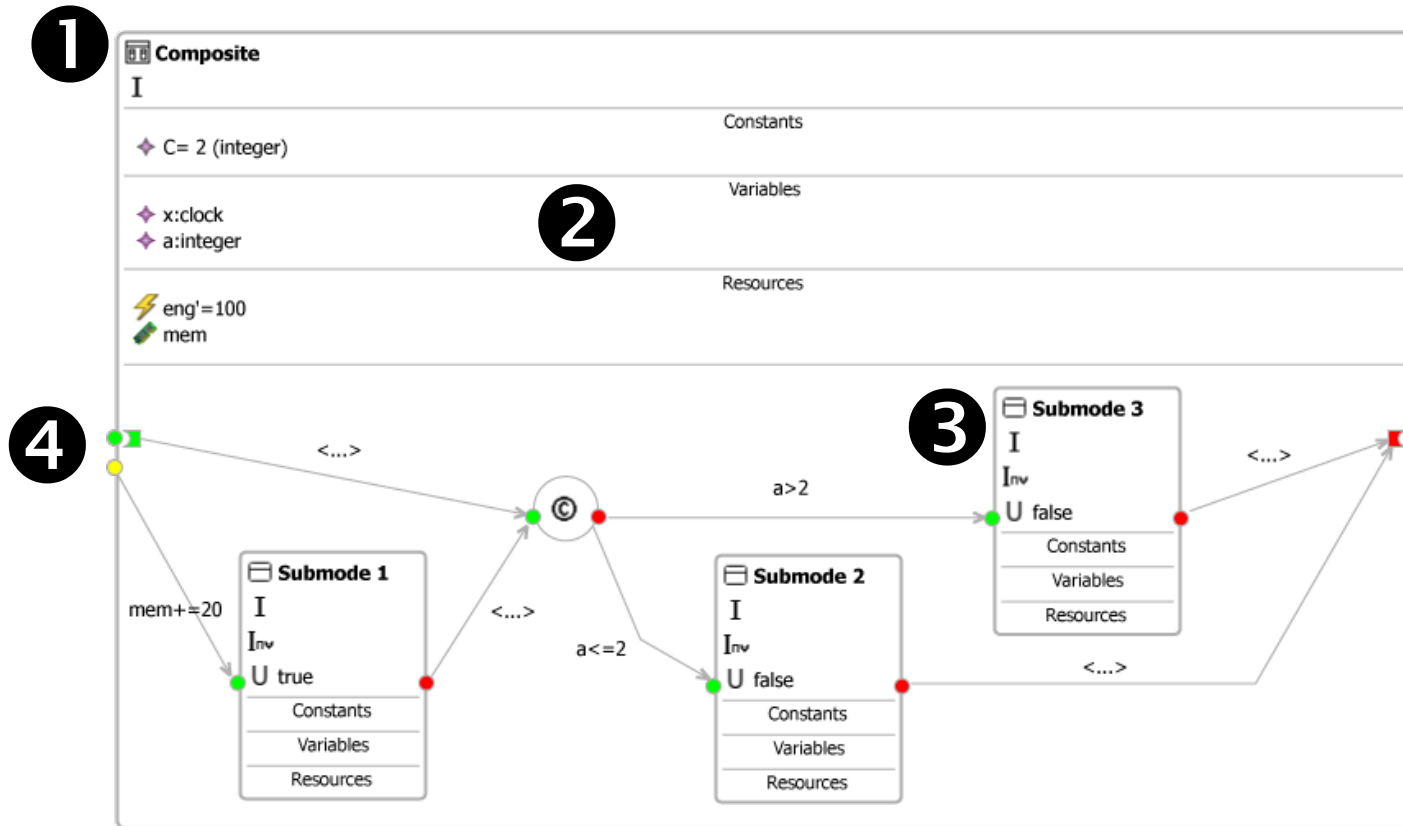
REMES language elements



- Submodes **3**

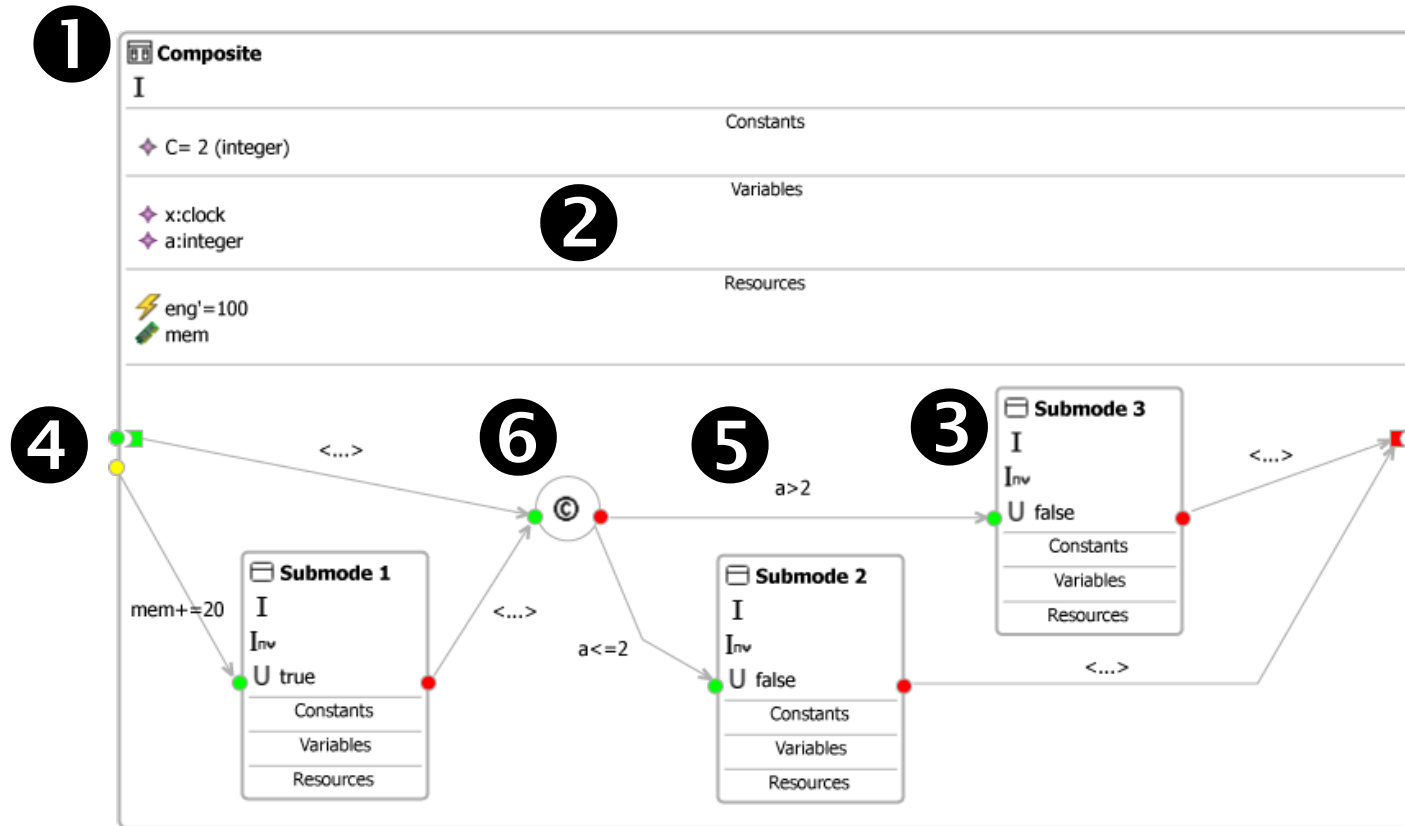
- Invariant – time is allowed to pass until invariant is violated
- Urgent – time is not allowed to pass (invariant is *false*)

REMES language elements



- Input and output ④
 - Init-, entry-, exit-, local exit points

REMES language elements



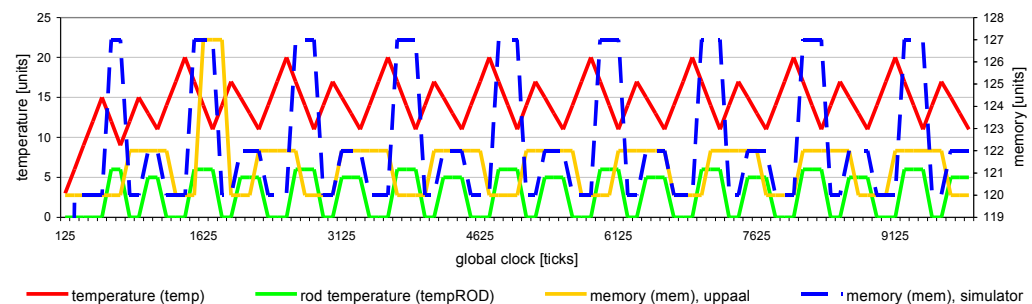
- Control flow
 - Edges with guards and actions **5**
 - Conditional connectors **6**

II. REMES simulator



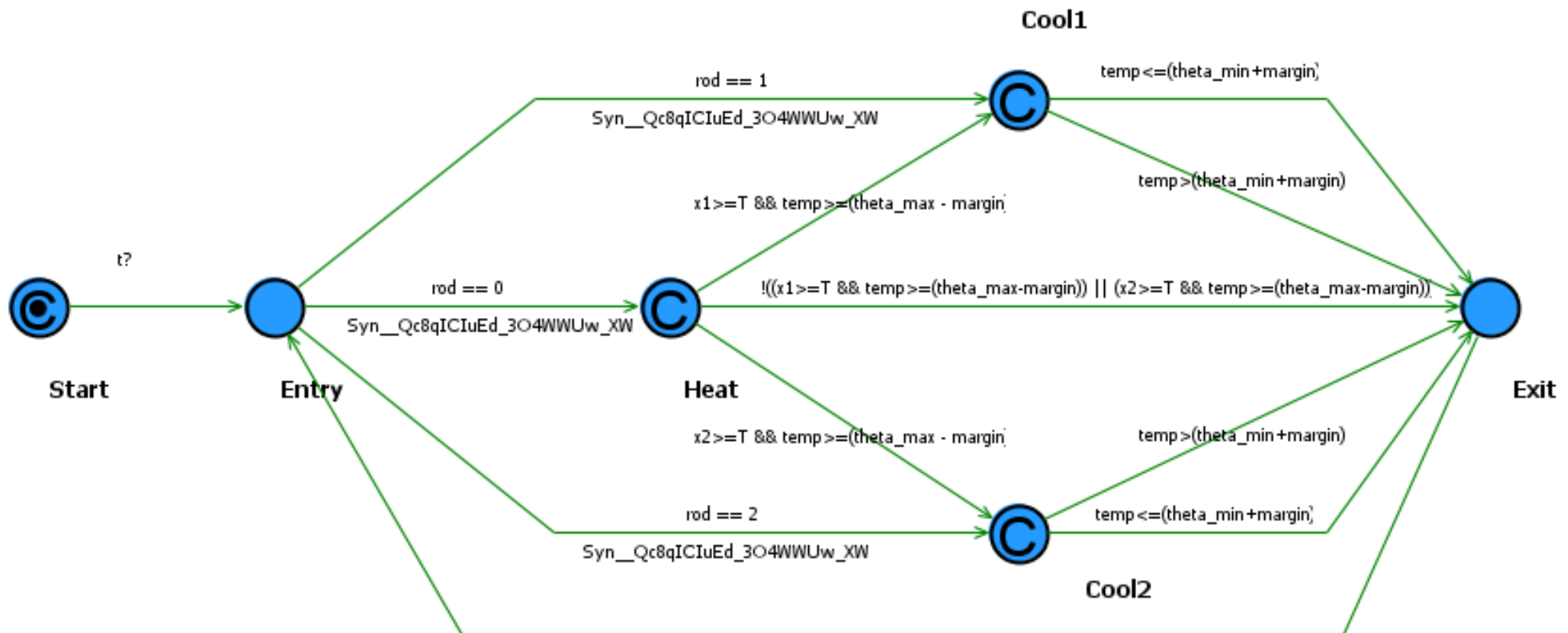
- Simulates the behaviour modelled in REMES
 - Single trace, out of all possible
- Output: mode transitions, clock- and variable changes
 - Quick prototype tests
 - Spot logical errors early
 - Perfect the model before formal analysis

```
<terminated> Sample [REMES behavior] C:\Program Files\Java\jre6\bin\javaw.exe (22. ruj. 2010. 16:19:25)
Entering Mode: SampleComposite
DEBUG: Earliest deadline: {0}, with 1 transitions
MULTIPLE OUTGOING TRANSITIONS ENABLED FOR SAME DEADLINE
Entering Mode: Sub1
DEBUG: Earliest deadline: {0}, with 2 transitions
Exiting Mode: Sub1
Entering Mode: Sub2
```



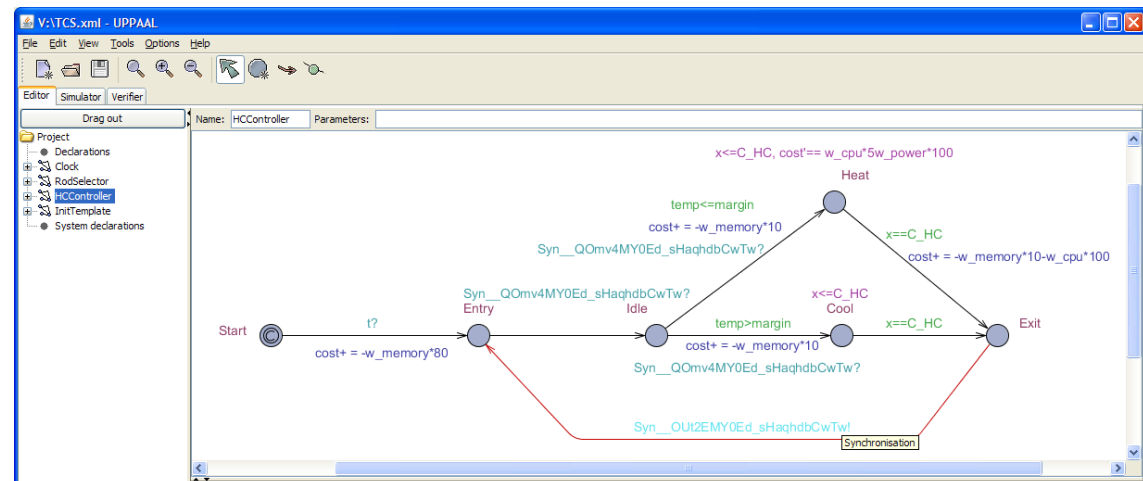
Analysis model

- Formal analysis – verify timing properties (TCTL)
- (Priced) timed automata – flat model



Formal analysis

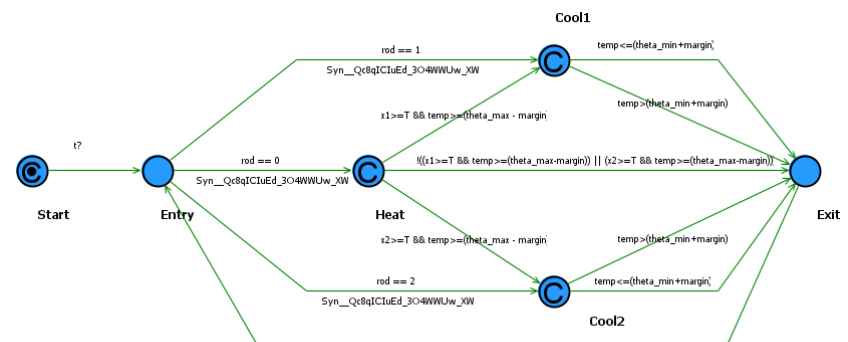
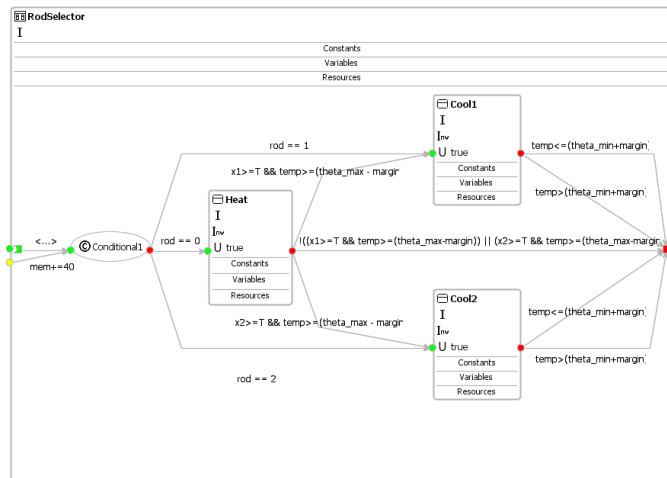
- Performed by tools of the UPPAAL family
 - Exhaustive search of the state-space, highest guarantee of model correctness
 - UPPAAL – timed automata
 - UPPAAL CORA – priced timed automata
 - Resources represented as a weighted sum in a single cost variable



III. REMES to PTA tool



- Automated transformation in IDE (M2M)
- Basic support for visual editing of PTA
- Export to UPPAAL (TA), UPPAAL CORA (PTA)
 - Includes triggering information (if available)
 - Integration with component model



IDE



ProCom - JavaProject/Sample.remes_diagram - PRIDE

File Edit Diagram Project Search Run Window Analysis Help

ProCom REMES Views Tahoma 9 B I A 100%

Project Explorer: JavaProject, src, JRE System Library [JavaSE-1.6], src-gen, REMES Simulator, system Models, Sample.remes, Sample.remes_diagram, Sample.uppaal, Sample.uppaalite, Sample.uppaalite_diagram, Sample2.uppaal, Sample3.uppaal, ProCom, JRE System Library [jre6], ProSave components, Clock, HCController, RodSelector, TCS, ProSave components/35b502dd-b7f..., ProSys components, REMES Simulator, .cp - D:\Dokume...Faks\PhD\PRID..., google-collect-1.0.jar - D:\Dokume..., system Models

Diagram: SampleComposite

Sub1: I, Inv, U false, Constants, Variables, Resources

Sub2: I, Inv, U false, Constants, Variables, Resources

Sub3: I, Inv, U false, Constants, Variables, Resources

Edges: $x \leq 2$, $x > 2$, $x <= 2$, $x > 2$, $x <= 2$

Diagram: LocalExit

Nodes: Start, Entry, Sub1, Sub2, Sub3, LocalExit

Edges: $x > 2$, $x <= 2$, $x > 2$, $x <= 2$, $t?$

Properties: Unassigned Attributes, Add Attribute

InitEdge

Core	Property	Value
	Connect To	◆ Sub1 Entry Point
	Initialization	■ mem+=50

Conclusions + future work



- Testing
 - Useful tool to follow timing and extra functional beh.
 - Ongoing work to test scalability on industrial case (Ericsson Nikola Tesla, Croatia)
- Integration
 - Integrate simulator, analysis model, integrate with PRIDE
- REMES updates
 - Implement all language improvements to tools
 - Support hierarchy
 - Provide feedback from formal analysis

Thank you

<http://www.fer.hr/dices/remes-ide>

Marin Orlić

marin.orlic@fer.hr

*Faculty of Electrical Engineering and Computing
University of Zagreb, Croatia*