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

Dinko Ivanov$^1$ Marin Orlić$^2$ Cristina Seceleanu$^3$
Aneta Vulgarakis$^3$

$^1$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
I. REMES language editor
REMES language elements

- Composite mode ①
- Compartments ② for variables, resources, constants
REMES language elements

- Submodes
  - 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
  - Init-, entry-, exit-, local exit points
REMES language elements

- Control flow
  - Edges with guards and actions
  - Conditional connectors
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
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
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