Introduction to Co-Simulation and the INTO-CPS Platform

Ken Pierce
Newcastle University, UK

18/04/2018
What is a Cyber-Physical System?

- Systems of interacting systems
  - Computing elements
  - Physical elements
  - Human interactions
- Complex, networked character
- Distributed control
- Error detection and recovery
CPS Engineering Needs

- Enable collaboration across disciplines
- Keep development costs low
- Keep time-to-market short
- Explore the complex design space efficiently
- Ensure tolerance against “nasty” faults
- Build up documentation for the working solution
- Provide confidence to external stakeholders
INTO-CPS Position

• Cyber-Physical Systems Engineering
  – The product is a system: software is not the end!
• Multidisciplinary collaborative modelling
• A tool chain, not single tools
  – Requirements and architectural models (in SysML)
  – Co-simulation
  – Traceability support through development
  – Test automation
  – Model checking
  – Hardware-in-the-loop (HiL) simulation
  – Design Space Exploration (DSE)
Model-Based Design

- Continuous-time (CT) models
  - High-fidelity models of physics
  - Differential equations
  - Poor support for supervisory controller design
- Discrete-event (DE) models
  - Supervisory control
  - Modal behaviours
  - Error detection and recovery
  - Poor support for physical modelling
Multi-Modelling and Co-simulation

- Combining CT and DE models into multi-models
- Analysis through co-simulation
- Whole-system models from heterogeneous parts
- Collaborating across disciplines

DE  CT  DE  CT

Co-simulation Engine

- Variables modified during run
- Design parameters fixed per run
- Shared design parameters and variables between models
- Ideal, realistic, faulty behaviours
Functional Mock-up Interface (FMI)

- Tool independent standard for co-simulation
- Models packaged as FMUs
  - Functional Mock-up Units
  - Black boxes to protect Intellectual Property (IP)
- XML describes an FMU interface
  - Called a model description

http://fmi-standard.org/
Maestro (Co-simulation Engine)

- Fully FMI 2.0 compliant Master Algorithm
- Multi-platform
  - 32-bit and 64-bit (Java-based)
  - Supports the most platforms
- Advanced features
  - Fixed- and variable-step
  - Constraints (zero-crossings; maximum / bounded difference; min. / max. sampling rates)
  - Parallelization (using Akka / Scala)
  - Nesting (multi-model as FMU)
  - Iterative solving (stability / stiff models)

https://github.com/INTO-CPS-Association/maestro
INTO-CPS Tool Chain

### INTO-CPS Application

- Configure / launch
  - Co-simulation
  - Design Space Exploration
  - Model Checking
  - Test Automation
- View Results
- Traceability

### Model Descriptions

- Overture
- 20-sim
- OpenModelica
- RT-Tester

### Modelio

- Export models
- Configures

### Traceability Daemon

- Traceability

### Application Flow

1. **Configure / launch**
   - Co-simulation
   - Design Space Exploration
   - Model Checking
   - Test Automation
2. **View Results**
3. **Traceability**

### CoE (Co-simulation Orchestration Engine)

- Code
- Export
- Co-simulate

- Generate Results
- HiL

- 20-sim 4C
INTO-CPS Application Demo
Design Space Exploration (DSE)

- Design decisions form a design space
- Use co-simulation to explore this space
- Help designers to trade-off between choices
Virtual Reality and Design Spaces
INTO-CPS Association

• Goals
  – Maintain and develop INTO-CPS further
  – Enable the widespread use INTO-CPS through open source licenses
  – Promote usage in academic and industrial settings
  – Allow members to steer direction of development and incorporate software into products

• Membership (in-kind or fees)
  – Associate (non-commercial; commercial)
  – Silver (associate + voting rights)
  – Gold (silver + IP protection)

http://into-cps.org/membership/
Introduction to Co-Simulation and the INTO-CPS Platform

Ken Pierce
Newcastle University, UK

18/04/2018