INTO-CPS: An well-founded integrated tool chain for comprehensive Model-Based Design of Cyber-Physical Systems

Professor Peter Gorm Larsen
Department of Engineering, Aarhus University
Head of Software Engineering

INTO-CPS

www.into-cps.au.dk
Who am I?

- Professor Peter Gorm Larsen; MSc, PhD
- 25+ years of professional experience
  - ½ year with Technical University of Denmark
  - 13 years with IFAD
  - 3,5 years with Systematic
  - 10 years with IHA/Aarhus University
- Reviewer for EU on Research projects and applications
- Consultant for most large defence contractors on large complex projects (e.g. Joint Strike Fighter)
- Mostly proud of the firmware of a NFC chip in 250+ million phones
- Relations to industry and academia all over the world
- Has written books and 100+ articles (in particular about VDM)
- See [http://pure.au.dk/portal/da/pgl@eng.au.dk](http://pure.au.dk/portal/da/pgl@eng.au.dk) for details
Short video overview of my own research
Background
• Embedded Systems
• Co-Modelling, Co-Simulation

INTO-CPS project
• Cyber Physical Systems (CPSs)
• System Vision
Embedded Systems

• Interacting computing, physical, human elements
• Increasingly complex logic (e.g. modeing) ~80% of control software
• Error detection and recovery

• Collaborative development
• Diverse disciplines cultures, abstractions, formalisms
• Typically tackled separately
• Need for design space exploration
Model-driven Design

• Modern systems are complex
• To cope with this, we can build models beforehand
  – To perform analysis (e.g. static analysis, proof, model checking, simulation)
  – Clarify our assumptions
  – Evaluate potential designs
  – Avoid expensive prototypes
• Different modelling paradigms for different aspects
Modelling of Software and Physics

- Typically **discrete-event (DE)**, e.g. VDM-RT based on discrete mathematics
- In simulation, only the points in time at which the state changes are represented
- Good abstractions for software,
  - e.g. data types, object-orientation, threading
- Less suited for physical system modelling

- Typically **continuous-time (CT)**, e.g. differential equations
- In simulation, the state changes continuously through time
- Abstractions for disciplines,
  - e.g. mechanical, electrical, hydraulic
- Poor software modelling support
  - only basic programming support; no functions or objects
Background: Co-modelling

Software:
- Discrete
- Complex logic

Physics:
- Continuous
- Numerical

Mind the Gap!
Background: Co-simulation

Discrete-Event Simulator → Co-Simulation Engine → Continuous-Time Solver
Co-simulation and real world
Reference Books

- Validated Designs for Object-oriented Systems
- Dynamical Systems for Creative Technology
- Collaborative Design for Embedded Systems

Baseline Discrete Event Modelling
Baseline Continuous Time Modelling
Co-Modelling
INTO-CPS: A new 8 M€ H2020 Project
Cyber-Physical Systems

- We have looked at individual embedded systems
- CPSs are networked groupings of digital devices
- ... which may require more elaborate co-models!
Cyber-Physical Systems

Control

Physics

Physics

Physics

Control

Physics

Physics

Control

Control
INTO-CPS

Strong Traceability
Configuration Management
1. Build an open, well-founded tool chain for multidisciplinary model-based design of CPS that covers the full development life cycle of CPS

2. Provide a sound semantic basis for the tool chain

3. Provide practical methods in the form of guidelines and patterns that support the tool chain

4. Demonstrate in an industrial setting the effectiveness of the methods and tools in a variety of application domains.

5. Form an INTO-CPS Association to ensure that project results extend beyond the life of the project
CPS co-modelling

requirements

architecture models

automated co-model analysis
design space exploration

models of cyber elements
- shared computing
- shared network

models of physical elements
- environment model

co-model

models of cyber elements

stub model generation

co-simulation (MiL)

SiL
HiL

models of physical elements

real plant

real code

code generation

labatory testing

realisation

model checking

analysis plug-ins

Andrey.Sadovykh@soTeam.fr
Combining Baseline Tools

Modelio \rightarrow \text{INTO-CPS CoSimulation} \rightarrow \text{RT Tester}

INTO-CPS CoSimulation

SYSML \rightarrow \text{FMI}

\text{FMI}

OpenModelica .fmu 20sim VDM
Industrial Follower Group

- AGCO, Denmark
- Alcatel-Lucent, Ireland
- Almende, Netherlands
- Altran, UK
- Bachmann electronic, Netherlands
- Bakker Sliedrecht Electro Industrie, Netherlands
- Bombardier, Germany
- Carrier, France
- CeTIM, Netherlands
- Chemring TS, UK
- Conpleks Innovation, Denmark
- Danish Aviation Systems ApS, Denmark
- DEME Group, Netherlands
- Denso Corporation, Japan
- Dredging International, Belgium
- DSTL, UK
- ESA, European Space Agency, Netherlands
- EDF, France
- Farmertronics BV, Netherlands
- Goodrich, UK
- Grundfos, Denmark
- GN Resound, Denmark
- HMF, Denmark
- Huisman Equipment, Netherlands
- Irmato Industrial Solutions, Netherlands
- Jaguar Land Rover, UK
- MAN Diesel & Turbo, Denmark
- Mfatech Limited, UK
- National Institute of Informatics, Japan
- ONERA, France
- Polar Electro, Switzerland
- Rockwell-Collins, France
- Rolls-Royce, UK
- Seluxit, Denmark
- Siemens, Sweden
- Terma, Denmark
- Thales, France
- TTTTech Computertechnik, Austria
- UTC Aerospace Systems, UK
- West Consulting, Netherlands
Initial Vision
The Initial INTO-CPS Vision

Design Space Exploration
Test Automation

Requirements

Feedback
MiL Co-Simulation
Testing

Heterogeneous Systems Models

SysML - FMI Model Generation

HiL / SiL Simulation

Code / Hardware

Strong Traceability
Configuration Management
Requirements Modelling

- **SysML**
  - Use Case diagrams
  - Requirements diagrams
    - Informal (link and traceable)
    - Formal (LTL, Test automation)

```
req FaultModeling
  «requirement»
  faultyBehaviour
  Model faulty behaviour in the sensors
  id=s1

«requirement»
ambientLight
Model ambient light as noise in the optical sensors
id=s1.1

«requirement»
conversionError
Model AD conversion errors in the LSB’s of the optical sensor readings
id=s1.2

«requirement»
sensorMalfunction
Model malfunctioning sensors that continuously reads the same value
id=s1.3
```

```
«requirement»
ambientLight
Model ambient light as noise in the optical sensors
id=s1.1

«block»
SensorRight
```

```
uc Line following robot

Line following Robot
  «include»
  Adjust motor control signals
  Controller
  Read optical sensor values
  «include»
  Measure optical reflection
  Sensors
  Read encoder ticks
  «include»
  Measure encoder ticks
  Encoders
  Wheels
```

```
pgl@eng.au.dk
```
The Initial INTO-CPS Vision

- Requirements
- Heterogeneous Systems Models
- Code / Hardware
- MiL Co-Simulation
- HiL / SiL Simulation
- Feedback
- SysML - FMI Model Generation

Strong Traceability
Configuration Management
System Decomposition

• Block Definition Diagram (top level)
System Interface Modelling

- Internal Block Diagram
  - Divide into CT/DE constituent models/systems/components
  - Define interfaces between different components
System Behaviour

• Parametric Diagram
  – Define continuous behaviour of CT components

• State Machines (DE models generated for tests)
  – Define discrete behaviour of DE components
The Initial INTO-CPS Vision

Design Space Exploration
Test Automation

Requirements -> Heterogeneous Systems Models
SysML - FMI Model Generation

Feedback

MiL Co-Simulation

Testing

Code / Hardware

HiL / SiL Simulation

Strong Traceability
Configuration Management
Co-model Development

• Model fidelity influence simulation speed
Methodology: Ideal -> Reality -> Faulty

• DE/CT/Contract(interface)-first

\[ \text{CT-first development} \]

\[ \text{Contract definition} \]

\[ \text{DE-only modelling} \]

\[ \text{Integration of initial co-model} \]
The Initial INTO-CPS Vision

- Requirements
- Heterogeneous Systems Models
- Code / Hardware
- MiL Co-Simulation
- MiL Co-Simulation Feedback
- HiL / SiL Simulation
- Testing
- Design Space Exploration
- Test Automation

SysML - FMI Model Generation

Strong Traceability Configuration Management
Co-model Traceability

• Trace of model artifacts
  – Can be accessed both from VCS and graphically
  – Show multiple models and their properties
  – When multiple possibilities exist, use Design Space Exploration experiment design
  – If component can be finite, use model checking partially automated from semantics of model
  – Trace of model results/evidence
The Initial INTO-CPS Vision

- **Design Space Exploration**
- **Test Automation**

**Requirements** → **Feedback** → **Heterogeneous Systems Models** → **MiL Co-Simulation** → **Testing** → **Code / Hardware** → **HiL / SiL Simulation** → **SysML - FMI Model Generation** → **Strong Traceability**

Configuration Management
Design Space Exploration

- Determine significant dimensions
- Design experiments
- Start sweeping to find optimum
- Determine fault tolerance
- Choose desired configurations
Co-model Development

• When experiments show the model is fit for purpose, create co-model for Design Space Exploration
• When experiments show the model is fit for purpose, start test automation
• When experiments show the model is ready, gradually incorporate SiL + HiL in simulator
• User able to get an overview of development and evidence produced (access from different tools)
Any questions?