Meta-models Combination for Reusing Verification Techniques

Hui ZHAO, Ludovic APVRILLE, Frédéric MALLET
ludovic.apvrille@telecom-paristech.fr

Modelsward’2019
CPS Modeling Problematics

Combining techniques

Combination Language

Conclusion

Context and Problematics

Institut Mines-Telecom

Meta-models Combination

Feb. 2019
(Multi-scale) CPS Modeling Approaches

(From: Multi-scale approach from mechatronic to Cyber-Physical Systems for the design of manufacturing systems, Penas et al., 2017)
Independent Techniques

Modeling approaches

- Engineering Modeling, ex: ARCADIA (Capella)
- Safety & Security Modeling, ex: TTool
- Architectural Modeling, ex: OSATE (AADL) and annex
- ... 

Verification approaches

- Timing, scheduling, ex: Cheddar
- Model checking, ex: UPPAAL, Proverif
- ...
Problems

- Proliferation of models
  - Different models of computation and communication
- Coherency between views

Our contribution: Efficiently combining existing modeling and verification approaches to (better) design CPS
Combination of different modeling and verification technologies using meta-models
Workflow (Cont.)

Metamodels of ARCADIA

Temporary combinational Metamodel

Transformation Rule LIB

Metamodels of AADL

1 conform to

corresponding

corresponding

conform to

corresponding

corresponding

Temporary AADL Models

Architectural + Timing

Design/Analysis

Simulate

Traceback

LIB

Import

Arcadia Models

Functional

Design/Analysis

M2

M1

Metamodels of AADL

correspond to

conform to

Metamodels of ARCADIA

Temporary combinational Metamodel

Import

Transformation Rule

LIB

Export

conform to

corresponding

conform to

corresponding

Legend

7/17 Feb. 2019

Institut Mines-Telecom

Meta-models Combination
Workflow (Cont.)

Arcadia Models

Functional Design/Analysis

M1

Import

Export

Temporary AADL Models

Architectural + Timing Design/Analysis

Legend

Metamodels of ARCADIA

Temporary combinational Metamodel

conform to

Metamodels of AADL

Imports

Transformation Rule

LIB

Import

Arcadia Models

Functional Design/Analysis

M2

M1
Workflow (Cont.)

Legend:
- conform to
- corresponding
- to be implemented

Simulation

Traceback

Schedule 1

Schedule 2
Combination Language: Typical Rules

$\Gamma PP \sim \langle feature \rangle : Type[data|event|dataevent]^+$
## Language Operators

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>⋁</td>
<td>Start of transformation Rule</td>
</tr>
<tr>
<td>;</td>
<td>End of rule</td>
</tr>
<tr>
<td>⇝</td>
<td>Transfer</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Parent node</td>
</tr>
<tr>
<td>{}</td>
<td>Attribute</td>
</tr>
<tr>
<td>[[]]</td>
<td>Optional element</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>{ }+</td>
<td>Attribute to be created</td>
</tr>
<tr>
<td>¬</td>
<td>Ignore</td>
</tr>
</tbody>
</table>
Arcadia Views

Functional View
Logical (software) components and their functional interrelation

Physical Architecture View
Physical (hardware) component’s relationship and their interconnection
## Combination Rules for Arcadia/AADL

<table>
<thead>
<tr>
<th>Arcadia</th>
<th>AADL</th>
<th>Additional (optional) attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical component container ($C_{comp}$)</td>
<td>System, Process</td>
<td>{Runtime_Protection[true</td>
</tr>
<tr>
<td>Function ($F_{un}$)</td>
<td>Abstract, Thread</td>
<td>{Dispatch_Protocol[Periodic</td>
</tr>
<tr>
<td>Port ($P_{ort}$)</td>
<td>Port</td>
<td>{Type[data</td>
</tr>
<tr>
<td>Functional Exchange ($E_{fun}$)</td>
<td>Connection</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Annex</td>
<td></td>
</tr>
<tr>
<td>Physical Node ($N_{ode}$)</td>
<td>Device, Memory, Processor, Bus</td>
<td></td>
</tr>
<tr>
<td>Physical Port ($PP$)</td>
<td>0</td>
<td>Type[abstract</td>
</tr>
<tr>
<td>Physical Link ($PL$)</td>
<td>Bus/BusAccess</td>
<td></td>
</tr>
</tbody>
</table>

- $C_{comp}$: Logical component container
- $F_{un}$: Function
- $P_{ort}$: Port
- $E_{fun}$: Functional Exchange
- $N_{ode}$: Physical Node
- $PP$: Physical Port
- $PL$: Physical Link
Train Traction Control System (AADL)

Use of OSATE for modeling and verification
Verification performed with Cheddar
Conclusion and Future Work

Achievements

- Method for combining modeling and verification techniques
- Definition of a combination language

Future work

- Full implementation of rules
- Automated backtracing of verification results