New results in Platform Modeling

Edited and presented by
Tivadar Szemethy
ISIS, Vanderbilt University
Model Transformation for Hard Real-Time Systems I.

Giotto → E-code mapping using UML-based metamodels and graph rewriting techniques

1) “Proof of concept” for a declarative approach using GME/GReAT
2) Part of the DSML → Platform → Analysis model mapping chain
Giotto
Tasks, drivers
Ports
Program: set of timing constraints

E-Code
Tasks, drivers, guards
Ports
Program: sequence of instructions

Model Transformation for Hard Real-Time Systems II.

Euclid’s GCD algorithm
1) choose \((a, b)\) s.t. \(a \geq b\)
2) if \((a = 0)\) GCD is in \(b\)
3) ensure that \(a \geq b\)
4) let \(a := a - b\)
5) goto 2)

GReAT
- UML Patterns
- Actions \((\text{match, create, delete})\)
- Attribute Mapping boxes

RuleBlocks
- Port bindings
- Initial matches

TaskA

TaskA

TaskB

TaskB

TaskB

0 100 200 300 400 500

frequencies → time slots

"New Results in Platform Modeling", Tivadar Szemethy
Chess Review, Nov. 21, 2005
The **Platform Modeling Language (PML)** is a declarative formalism to capture platform entity → analysis automaton structure mappings.

Each Block has one Filter condition. “Downstream” elements refer to the objects matched by the Filter.

**Hierarchical Blocks**

**Filter Conditions**

**Associations**

**Actions**

**PML Semantics**

While \( \exists i \text{ such that } GFC_i \text{ is true} \)

Execute Action\(_i\)

**GFC\(_i\) (Global Filter Condition)**

Chain of filter conditions for Action\(_i\)

**PML Example**

DFK (Dataflow Kernel) to IF (timed FSM) mapping