Graph Transformation as a Meta Language for Dynamic Modeling and Model Evolution

Reiko Heckel and Gregor Engels
University of Paderborn
Germany
Summary

- Model–oriented classification (2 dimensions)
  - Layer of hierarchy
    - objects – SW components – HW components
  - Level of typing
    - class – instance

- Formalization
  - Meta modeling
  - Graph transformation
Meta Model: Hierarchy and Typing

hardware architecture

software architecture

object layer

Component instOf ComponentInstance

Class instOf Object

Node instOf NodelInstance

type level

instance level
Example: Type Level Hierarchy

- A *CashBox node* may host a *Billing component* for creating (and storing) *Bill objects*.

- A *SmartCard node* containing a *BillCard component* accepts a *Bill object* from a *CashBox*.
Example: (Type and) Instance Level

:Billing

   b :Bill
   amount = a

:BillCard

   b :Bill
   amount = a

:ComponentInstance

instOf

:ComponentInstance

instOf

:Object

instOf

Billing:Component

BillCard:Component

instOf

Bill:Class
Example: Type Level Transformation Rule
Example: Instance Level Transformation Rule

1. :SmartCard
2. installService
3. :CashCard

CashCard:Component

SmartCard:Node

:NodeInstance

instOf

CashCard:Component

:ComponentInstance

SmartCard:Node

:NodeInstance

instOf

instOf
Summary

- Model–oriented classification (2 dimensions)
  - Layer of hierarchy
    objects – SW components – HW components
  - Level of typing
    class – instance

- Formalization
  - Meta modeling
  - Graph transformation