Karlsruhe Institute of Technology

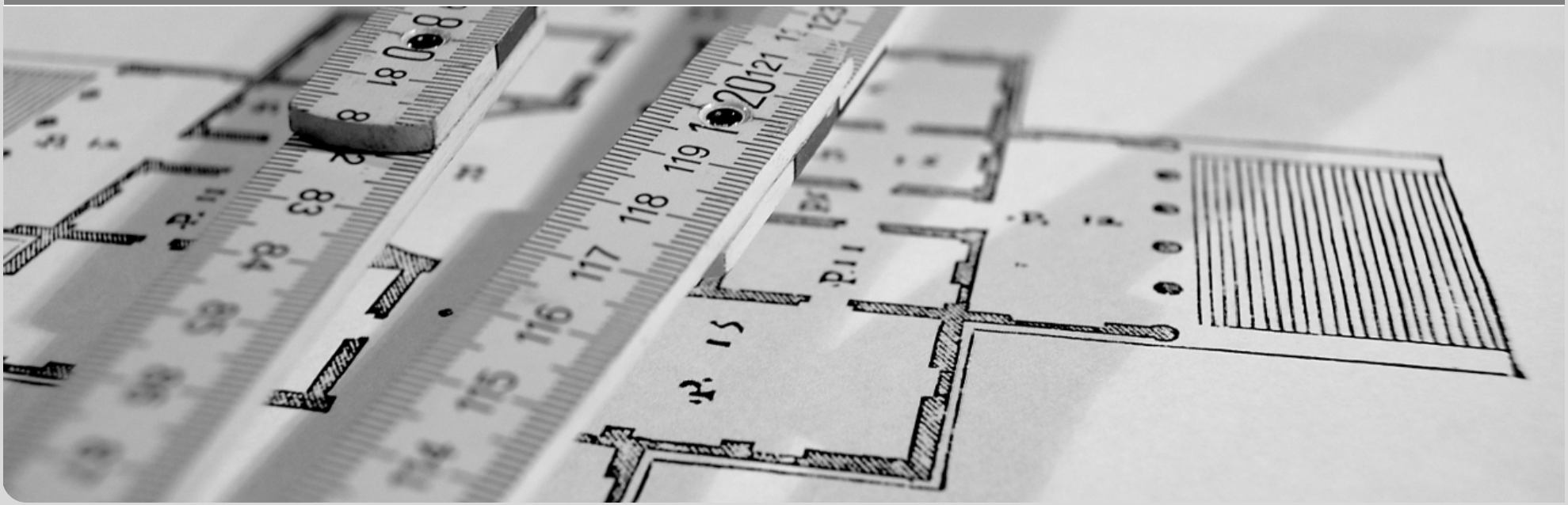
Software Design and Quality

Special Interest Group: Palladio Simulation Core

Palladio Days 2010

Jörg Henß | November 25, 2010

CHAIR FOR SOFTWARE DESIGN AND QUALITY



Why a Core Model?

- Reuse of models in several tools
  - Don't reinvent the wheel
  - Knowledge transfer
  - Eases concurrent development
- High degree of abstraction
  - Custom extensions can only be decorated
  - Core meant to be stable

Disambiguation - Core

- Core already used in different contexts
  - Intermediate Core: Interface to different prediction models
  - Advanced Model + Quality Decorator - PCM today + Decorator
  - Basic Core: Structure and Communication
- Core Model presented targeted at performance simulation
  - Model not intended to be modeled by “hand”
  - Intermediate model $\rightarrow$ target of transformations

Ideas for better Naming

- Minimal Set of Constructs that Are Behaviour Equivalent to the RDSEFF: MiSCABE
- Model Elements Representing Core Intermediates : MERCI

Goals of the SIG

- Creation of a core model for simulation purposes
  - Especially “full” discrete event-driven simulation
- Enhance portability of simulation models
  - Support specialised simulation plattforms
    - Network
    - Reliability
    - Reconfiguration
    - ...
- Ease the integration of Schedulers
  - Modeling of processes and threads

Intermediate Simulation Core Model

- Current work: Behavioural information
  - Reduction of (RD)SEFF to basic control flow instructions
  - Comparable to the RISC approach
  - Explicit semantics of each step
  - Adapted to fit event paradigm
  - Asynchronous communication
  - QoS Annotations
- Structural information
  - Components
  - Active and Passive Resources
  - Network topology (new)
- More detailed extensions can be annotated
  - E.g. network model

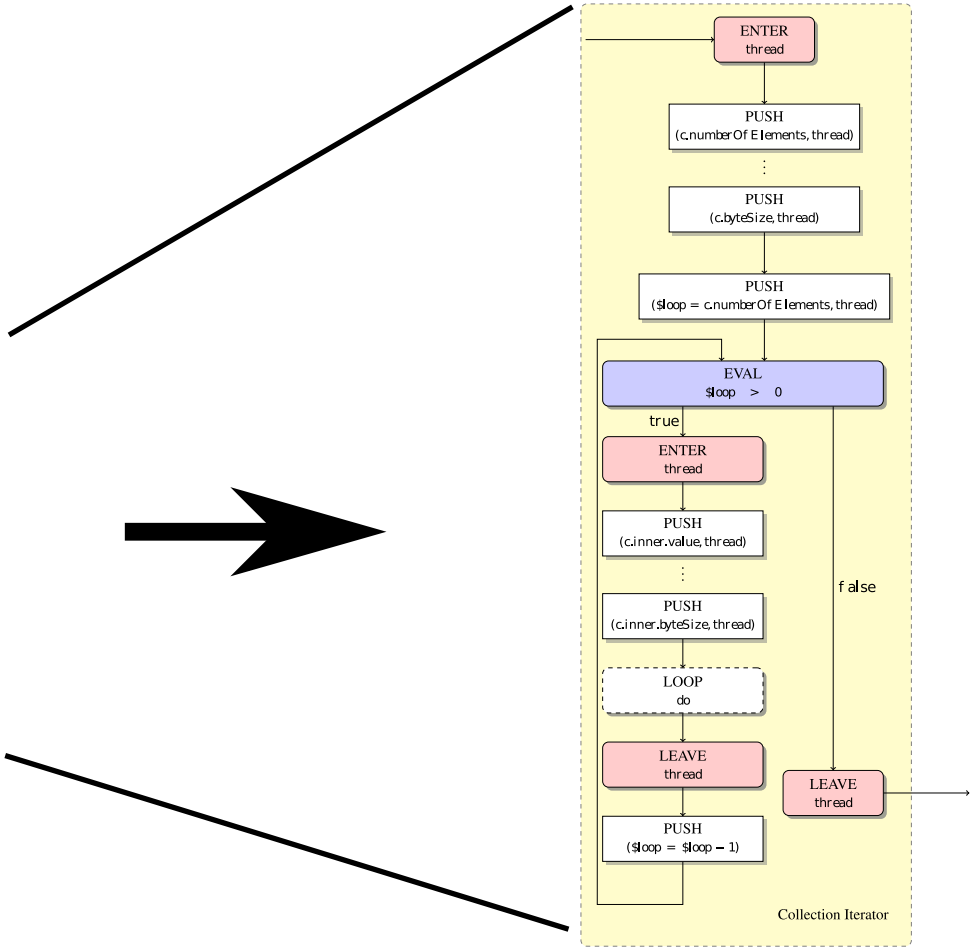
The RISC approach

- Reduction to basic instructions
  - Complex instructions are converted to sequence of basic instructions
  - $\rightarrow$ Less instruction have to be implemented
  - $\rightarrow$ Gain of expressiveness?!
- Planned Model Elements:
  - Variable Declaration and Assignment: Push, Context
  - Flow Control Statements: Eval, (Goto)
  - Resource Operations: ConsumeResource, Acquire, Release
  - Calling Operations: ReceiveRequest, SendRequest
  - System Operations: CreateThread/Process, Fork

Explicit Simulation Semantics

- Especially scopes of variables often unclear
  - Handling of variable scope solely dependent on SimuCom implementation / transformation
  - Include this implicit knowledge in the model
  - Closure like semantics
- How are request handled
  - Currently new process is spawned
  - Usage of interaction pattern
  - Support of threads and processes

Intermediate Simulation Model



Exemplary Model-driven Transformation

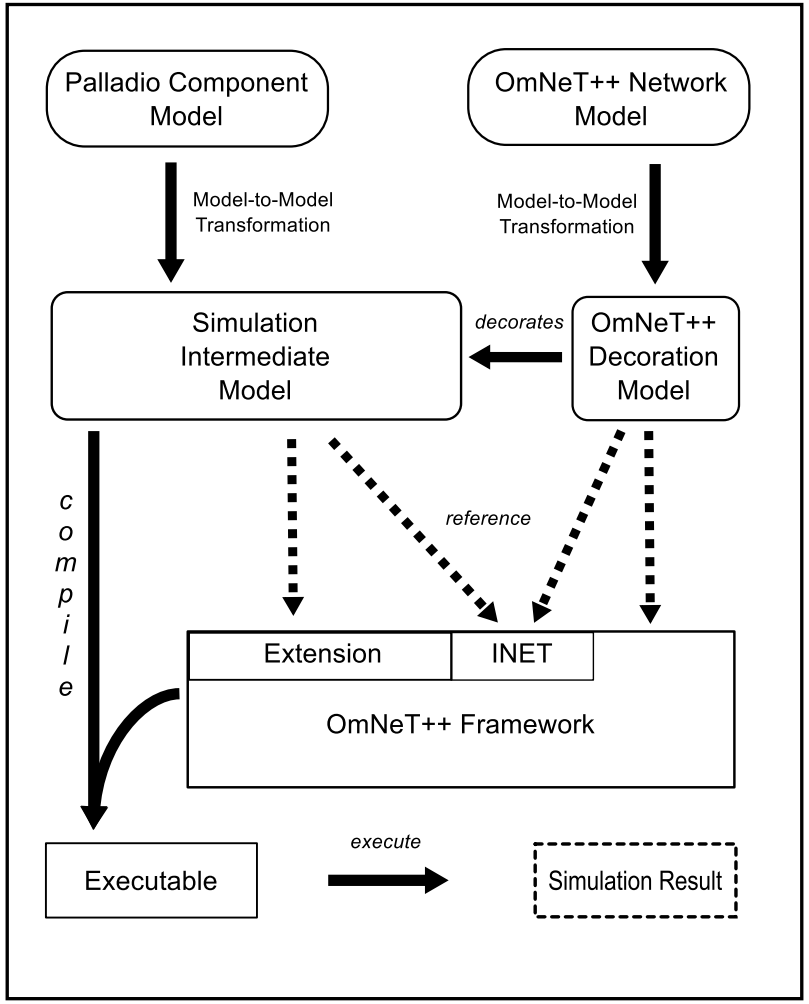


Figure: Schematic illustration of our integrated simulation approach

State of Work

- Conceptual behavioral model
  - Ideas documented
  - EMF model required for transformation → Q1 2011
- Implementation:
  - StoEx evaluation ported to C++
  - Prototype integration with OmNeT++ in progress
  - Master thesis bootstrapped for full event based SimuCom
    - Will evaluate parts of the approache
- Static structure planed to be rolled out

Open Questions - Variable Usage

- Currently kind of closure semantics is used
- Values are diced (almost) every time
- Ideas:
  - Storage of values through the control flow
  - Allow changes to variables $\rightarrow$ access modifier
  - Allow kind of communication through variables
  - Process, thread, component, system wide state

Open Questions

- Selection of model elements
  - Is GOTO semantics enough
  - Do we need additional elements
- How are systems “bootstrapped”
  - New process for every request not realistic?!
- Do we need more synchronisation primitives
  - Join threads (limited support today), “synchronized”, . . .
- Is this approach really more scalable?

Conclusion

- PCM Simultion Core still in development
- Reduced set of building blocks
- More expressiveness $\rightarrow$ more problems?
- Many open questions

Thank you for your attention!

References