Simulation Modeling of UML Software Architectures

Simonetta Balsamo, Moreno Marzolla

Dipartimento di Informatica, Università Ca' Foscari di Venezia
{balsamo, marzolla}@dsi.unive.it
Software Performance Modeling

- Early identification of performance problems in Software Architectures is very useful
  - Changing the design costs more if done late
- Two approaches to SA performance evaluation
  - Measurement-based
    - require a running system
  - Model-based
    - can be done at the design stage
Model-Based Approach

Software Architecture

Software Model

Performance Model

Performance Results

Perf. Model Evaluation

Feedback

UML-Ψ
Simulation for Software Performance Evaluation

- Simulation often considered a solution technique for analytical performance models
- Our approach: use simulation as the performance model
- Advantages:
  - Mapping between software model and performance model very easy
  - Can represent general software models
  - Easy to report feedback
UML

- UML is a standard notation for high-level software description
- Different kinds of diagrams available
- UML-Ψ uses two kinds of diagrams:
  - Use Case Diagrams
  - Activity Diagrams
The Simulation (Meta)Model

![Diagram of the Simulation (Meta)Model]

- **PerformanceContext**
- **Workload** with attributes:
  - population : Integer
  - externalDelay : PAperfValue
- **ClosedWorkload** with attributes:
  - population : Integer
  - externalDelay : PAperfValue
- **OpenWorkload** with attributes:
  - occurrencePattern : RTarrivalPattern
- **PScenario** with attributes:
  - root
  - steps
- **PResource**
- **AbsStep** with attributes:
  - probability : Double
  - repetition : Integer
  - delay : PAperfValue
  - interval : PAperfValue
  - PDemand : PAperfValue
  - responseTime : confInterval
- **PStep**
- **Pstep_fork**
- **PStep_join**
  + _successors
  + _predecessors
Use Case Diagrams

- Actors are used to represent workloads applied to the system.
- Each new user performs one of the associated Use Cases.
Activity Diagrams

- Each Use Case is expanded into a number of activities

Browse Catalog

Make Order

- Issue Request
  - PAdemand=["assm","dist", ["exponential",1.0/2.0]]
- Compose Page
  - PAdemand=["assm","dist", ["exponential",1.0/1.0]]
- Select Product
  - PAdemand=["assm","dist", ["exponential",1.0/15.0]]
  - PAdelay=["assm","dist", ["exponential",1.0/35.0]]
- Fill Order Form
- Process Order
- Verify Payment
  - PAdemand=["assm","dist", ["exponential",1.0/2.0]]
How UML-Ψ works / 1

- The starting point is a set of UML Use Case and Activity diagrams.
  - We use ArgoUML as a graphical tool for manipulating UML diagrams
- UML diagrams are exported in XMI format
- UML-Ψ parses the XMI file, building the simulation model
  - Actors  →  Workload
  - Use Cases  →  PScenario
  - Activities  →  AbsStep
How UML-$\Psi$ works / 2

- Annotations define parameters of the simulation model
- The simulation model is executed
  - Based on a custom process-oriented, discrete-event C++ simulation library, providing SIMULA-like process scheduling facilities
- Simulation computes the average delays of Activities and Use Cases execution
- Results are put into the original XMI file as tags associated to the appropriate UML element
Conclusions

- We proposed a simulation-based performance modeling approach for UML software architectures
- UML annotations based on a subset of the UML Performance Profile
- Simulation model is implemented in C++
- Feedback reported at the UML level
Future Work

- We are currently working on resource modeling in a UML context using Deployment diagrams
  - Active resources have been implemented
  - Passive resources still work in progress
- We are also extending the approach to performance evaluation of mobile systems
  - No “standard” way to represent mobility in UML