fUML-Driven Performance Analysis through the MOSES Model Library

Luca Berardinelli, Vittorio Cortellessa
Post Doc @ DISIM
University of L’Aquila

*ACES-MB Workshop@MODELS2014
MOSES and PRESTO

MOSES: **MOdeling Software and platform architecture in UML 2 for Simulation-based performance analysis**

**Scope:**

- Software Architecture
- & Hardware Platform **Modeling**
- Non-Functional Validation
- Performance Analysis

[Link to website: http://www.presto-embedded.eu/]
MOSES in a Nutshell

MOSES: MOdeling Software and platform architecture in UML 2 for Simulation-based performance analysis

Everything is a fUML Model

System Model
- Software Architecture
- Hardware Platform

MOSES
- Model Library
- Performance Analysis Results
MOSES in a Nutshell

Everything is a fUML Model.

System Model
- Software Architecture
- Hardware Platform

<<extended by>>

MOSES
- Model Library
- Perf. Analysis Results

<<generates>>
fUML Virtual Machine

<<simulated by>>

Execution Model
MOSES in a Nutshell (tools)

Everything is a fUML Model

System Model

Software Architecture
Hardware Platform

MOSES
Model Library
Perf. Analysis Results

fUMLVirtual Machine

<<extended by>>

<<generates>>

<<simulated by>>
Foundational UML

Everything is a... lightweight extensions by

<table>
<thead>
<tr>
<th>COMPOSITE STRUCTURES</th>
<th>COMPONENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPLOYMENTS</td>
<td>INTERACTIONS</td>
</tr>
<tr>
<td>STATE MACHINES</td>
<td>USE CASES</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Luca Berardinelli, Philip Langer, Tanja Mayerhofer: **Combining fUML and profiles for non-functional analysis based on model execution traces.** QoSA 2013: 79-88

**Formalizing Execution Semantics of UML Profiles with fUML Models (F)**
Jérémie Tatibouet, CEA, France
Arnaud Cucuru, CEA, France
Sébastien Gérard, CEA, France
François Terrier, CEA, France
Foundational UML: Generic Example

Everything is a fUML Model

Class Diagram(s) + Activity Diagram(s)
Case Study: Indoor Positioning System

Everything is a fUML Model

System Model
Indoor Positioning System

Software Architecture
OLSR protocol

Hardware Platform
MANET Nodes
Case Study: Indoor Positioning System

- **Indoor Positioning System (IPS)** based on a Mobile Ad Hoc NETwork (MANET), nodes are mobile, connected by wireless links

- **Software (COTS):**
  - Optimized Link State Routing (OLSR), IP routing
  - **Standard**: RFC3626, defines data structures (tables) and functionalities
  - **Implementation**: OLSR Daemon, olsrd

- **Hardware:**
  - **ATMEL ATZB-900-B0** module that sustains, with its transceiver, the **signaling** among nodes
  - **OMAP L138** module sustains **physical, medium access control**, and **network layers** of the communication network
Indoor Positioning System (structure)
Indoor Positioning System (behavior)

Call Operation Action to...

OLSR functionality

LinkSet::update()
Everything is a fUML Model

The MOSES Model Library

System Model
Indoor Positioning System

Software Architecture
OLSR protocol

an intermediate, executable layer
to carry out performance analysis of a
System Model
on top of fUML VM simulation

Hardware Platform
MANET Node(s)

MOSES

Model Library

Performance Analysis Results
MOSES (structure, informal)

System Model
Indoor Positioning System

Software Request(s)
(invoking OLSR functionalities)

Resource Request(s), 1..*
(invoking Components' operations)

MOSES

Workload Generator

Software Architecture
OLSR protocol

Main Dispatcher

Computing Request(s), 1..*
Communication Request(s), 0..*
Storage Request(s), 0..*

CPU Dispatcher

Network Dispatcher

Disk Dispatcher

Computing Resources
FCFS CPU
RR CPU
...

Communication Resources
WIFI
UMTS
...

Storage Resources
ROM
RAM
...

Hardware Platform
MANET Node(s)
MOSES (behavior, informal)

System Model
Indoor Positioning System

additional data:
timings for execution occurrences
MOSES (structure, fUML)

System Model
Indoor Positioning System

Software Architecture

Workload Generator
connected to software components that receive Software Requests
MOSES (structure, fUML)

System Model
Indoor Positioning System

timings for execution occurrences (e.g., arrival time, service time)
generated by random number generators
(e.g., waiting time, completion time)
derived slot values by simulation of activities
MOSES (structure, fUML)

System Model
Indoor Positioning System

Simulation Parameters:
e.g., max number of Software Request generated by the Workload Generator

Simulation Results:
e.g., System Response Time, CPU Utilisation

Hardware Platform
MANET Node(s)
Two main steps:
- Generalizations
- Extension of Software Behaviors with MOSES-specific actions
• An IPS's Node is the MOSES Model we are going to simulate a single node rather than the whole network.

• OLSR HELLO Message is the Software Request.

• OLSR is a SwComponent.

• OLSR's Repositories are SwComponents.

• OMAP L-138's CPU is a FCFS CPU.
OLSR Functionality: HELLO Message Processing

model instrumentaYon
LinkSet::update()
addi$ional	 data:
$mings
for  execu,on
occurrences
OLSR	 FuncYonality:
HELLO	 Message	 Processing
LinkSet::update()
model instrumentation
LinkSet::update()
additional data: 
timings
for execution occurrences
Performance Analysis by fUML VM

Everything is a fUML Model

System Model- Indoor Positioning System
- **Workload**: HELLO Message(s)
- **Processing Steps**: Flooding of HELLOs
- **Timings** for ALL the involved components' operations.

MOSES
- Model Library
- Perf. Analysis Results

GUI

fUMLVirtual Machine

Execution Model

<<extended by>>

<<generates>>

<<simulated by>>
Performance Analysis by fUML VM

Workload Generator Parameters: arrival pattern of HELLOs

Stop Conditions: e.g. MAX #HELLOs

Operations' Resource Demands

Performance Analysis Results
Performance Analysis by fUML VM

Assumed or Measured Parameters
- Service Time per Operations
- Exponentially Distributed Variables

"Time" is not natively managed by the fUML VM
Performance Analysis by fUML VM

Table 1: Timing parameters for components’ operations.

<table>
<thead>
<tr>
<th>Component::Operation</th>
<th>$Avg(ST)_{1SR}$ (ms)</th>
<th>$Avg(ST)_{25SR}$ (ms)</th>
<th>$\lambda_{25} = (1/Avg(ST)_{25SR})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinkSet::update()</td>
<td>149.00</td>
<td>3725.00</td>
<td>$3.0E-4$</td>
</tr>
<tr>
<td>NeighborSet::update()</td>
<td>15.86</td>
<td>396.50</td>
<td>$2.5E-3$</td>
</tr>
<tr>
<td>MPRSelectorSet::update()</td>
<td>97.96</td>
<td>2449.00</td>
<td>$4.0E-4$</td>
</tr>
<tr>
<td>2HopNeighborSet::update()</td>
<td>1.54</td>
<td>38.50</td>
<td>$2.6E-2$</td>
</tr>
<tr>
<td>MPRSet::recalculation()</td>
<td>1.72</td>
<td>43.00</td>
<td>$2.32E-2$</td>
</tr>
</tbody>
</table>

Table 2: Performance Analysis Results.

<table>
<thead>
<tr>
<th>Performance Index</th>
<th>Required</th>
<th>Estimated</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Utilization (%)</td>
<td>&lt; 40</td>
<td>11.8</td>
</tr>
<tr>
<td>Max System Response Time (single HELLO - batch of 25 HELLOs) (ms)</td>
<td>80 - 2000</td>
<td>21.19 - 529.74</td>
</tr>
</tbody>
</table>
Future Work (1)

- Extending **analysis capabilities**
- Enhancing its **design** (it is a model!)

implementation biases w.r.t. the chosen fUML VM (Cameo Simulation Toolkit)?

Luca Berardinelli, Antinisca Di Marco, Stefano Pace:
ECSA 2014: 324-339
Future Work (2)

- Applying fUML-driven analysis in **different domains** (e.g., Cloud-based Systems)

System Model

- Software Architecture
- Hardware Platform

fUML Virtual Machine

<<simulated by>>

Perf. Analysis Results

<<generates>>

Model Library

Martin Fleck, Luca Berardinelli, Philip Langer, Tanja Mayerhofer, Vittorio Cortellessa: 
**Resource Contention Analysis of Cloud-based System through fUML-driven Model Execution.**
NiM-ALP@MoDELS 2013: 6-15
• Studying the suitability of the current of the fUML VM for Non-Functional Analyses
  Experienced scalability problem with large workloads

http://www.modelexecution.org/
fUML-Driven Performance Analysis through the MOSES Model Library

Thank You

Luca Berardinelli, Post Doc @ DISIM
University of L’Aquila