Introduction:

Introduction

Michael Barth
Research Group on Programming and Software Engineering
(Professor Martin Wirsing)
Department of Computer Science
University of Munich, Germany
(Ludwig-Maximilians-Universität München)
Overview

- How do we use the UML notation.
- How to extend the resource model packages.
- How to generate resource models by measurements.
- How to evaluate models with the simulator.
- Examples and add ons
Approach to Performance Modeling

- UML Diagrams: activity diagrams, class diagrams.
- High level of acceptance by using simple modeling techniques.
- Standard tools are used to edit the diagrams.
- Simulators and tools processing XMI files.
- Dynamic models that are intuitively understandable.
- Using one technique throughout the whole process.
The Simulator (Overview)
UML Diagrams:

UML Diagrams

- Dynamic models are described by UML activity diagrams.
- Data models are described by object flow states in activity diagrams.
- The environment is described by a class diagram.
Model Elements (Overview)

- Activity Models
- Data Models
- Resource Models
- Dynamic SW Model (Activity Diagram)
- Environment Model (Class Diagram)
Control Flow Modeling

Execution of a single activity:

```
entry / client.execute(new SQLQueryExecution())
```

In this example a client is executing a SQL query.
Data Flow Modeling

A data object is generated by activity ’A’ and consumed by activity ’B’.
Environment Modeling

The environment is modeled in a class diagram.
Extension of Resource Package Classes

The resource package can be extended by:

- Modified resources
- Combination of resource classes
- Extension of abstract resources
- Generated classes from measurements
Modified Resources

\[
\begin{array}{|c|}
\hline
\text{hera : PC} \\
\hline
\text{memorySize} = 256 \\
\text{clock} = 1500 \\
\text{diskSize} = 32000 \\
\text{diskSpeed} = 0.07 \\
\hline
\end{array}
\]

Values of attributes can be modified in the class diagram of the environment description.
Combination of Resource Classes

Resources consisting of several basic components can be variedly composed.
Extension of Abstract Resources

Abstract resources can be extended by implementing a method to calculate the load function.
Model Generation

- Select required activity models.
- Develop an implementation example.
- Wrap the example in a Java method.
- Run the method on a resource.
- Create a measurement table from the test run.
- Create a resource model class from the test run table.
Model Generation

Model vs. Reality

Activity Model

\[ \text{describes} \]

Class

\[ \text{method()}:\text{type} \]

\[ \text{executes} \]

Resource Model

\[ \text{describes} \]

Real Resource

executes
Model Generation Process

C++ Testimplementation
Java Wrapper
attached to
creates
Survey Table
Resource executes
Monitoring Tool
«abstract» ResourceModel
Model Generator
ConcreteModel

C++
Testimplementation
Java Wrapper
attached to
creates
Survey Table
Resource executes
Monitoring Tool
«abstract» ResourceModel
Model Generator
ConcreteModel
Model Evaluation:

Model Evaluation

The following steps are to be done to evaluate a model:

- The virtual environment is compiled from the environment description.
- The dynamic model is compiled from the activity diagram.
- In a specification a single scenario is determined.
- The simulator evaluates the processing time or processing function.
- The Result is displayed by the simulator.
Results

Performance Analyses: SendForm.AD [DemoMdl.xmi]

ResponseTime

DataForm.size

Scenario Specification:
Testrange: Parameter: From: To:
6fk*t: DataForm.size 1 20000
knottable.def

Fixed Values: ScenSpec-aad03.def
Testrun-ID: CD030412.0023
Add Ons:

Add Ons

- Stereotype 'create' allows to model persistent memory consumption.
- Stereotype 'consume' allows to model release of memory.
- Stereotype 'persistent' models memory consumption if objects that are not consumed during an execution.
Persistent Data

Activity A

<<create>>

<<persistent>>

data: Object

Activity B

<<consume>>

Activity C
Conclusions:

Conclusions

- Intuitively understandable models.
- Applicable in each phase of a project.
- Models can be developed with standard tools.
- Extension mechanisms do allow one to proceed in a top down manner.
- Many other (more sophisticated) performance model can be integrated in the future.
Conclusions: Future Work

Future Work

- Enhanced monitoring tool.
- Integration of other performance models.
- Development of resource model that do have states.
- Enhanced activity packages and extension mechanisms.
- Development of a methodology.