Business Informatics Group

Vienna University of Technology

A Runtime Model for fUML

MRT 2012 – 7th International Workshop on Models@run.time

www.modelexecution.org

Tanja Mayerhofer, Philip Langer, Gerti Kappel

Business Informatics Group

Institute of Software Technology and Interactive Systems
Vienna University of Technology

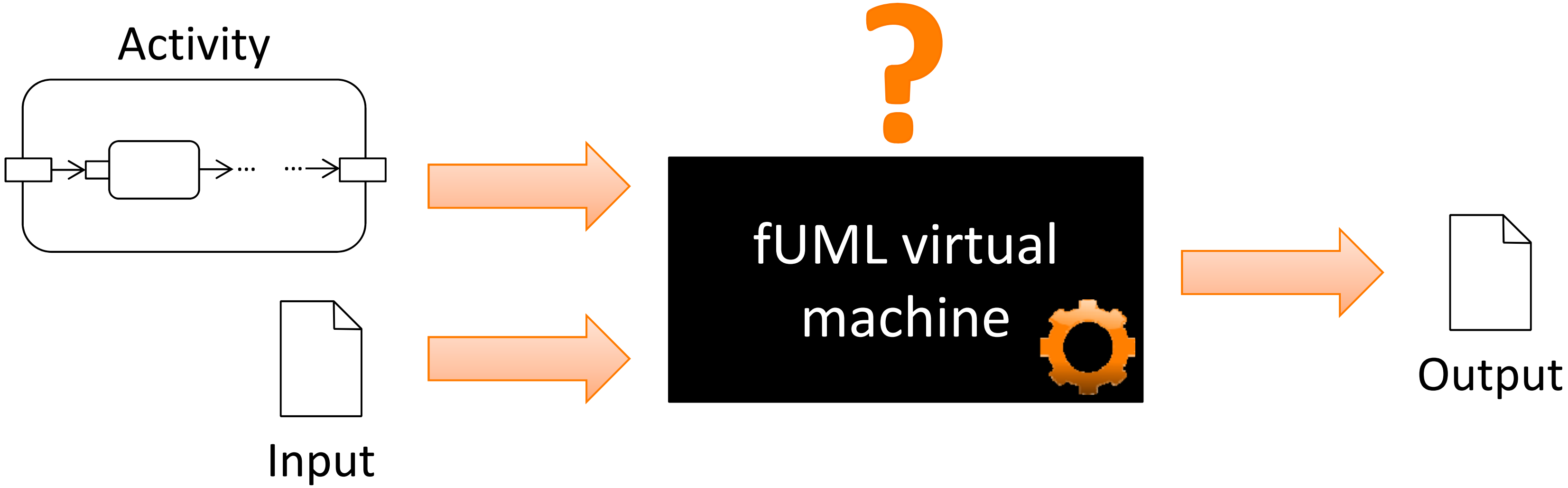
Favoritenstraße 9-11/188-3, 1040 Vienna, Austria
phone: +43 (1) 58801-18804 (secretary), fax: +43 (1) 58801-18896
office@big.tuwien.ac.at, www.big.tuwien.ac.at

Introduction

Foundational UML (fUML)

- OMG standard V1.0 released 02/2011
- Specification of dynamic semantics for UML class diagram and activity diagram
- Operational semantics approach specifying a UML virtual machine
- UML evolves to implementation language

Problem Description



- ? Which activity nodes have been executed?
- ? In which order have the nodes been executed?
- ? Which values have been consumed as input / produced as output by a node?
- ? What other activities have been called?
- ? What has the input / output of a called activity been?

Contributions

fUML virtual machine lacks in providing means for

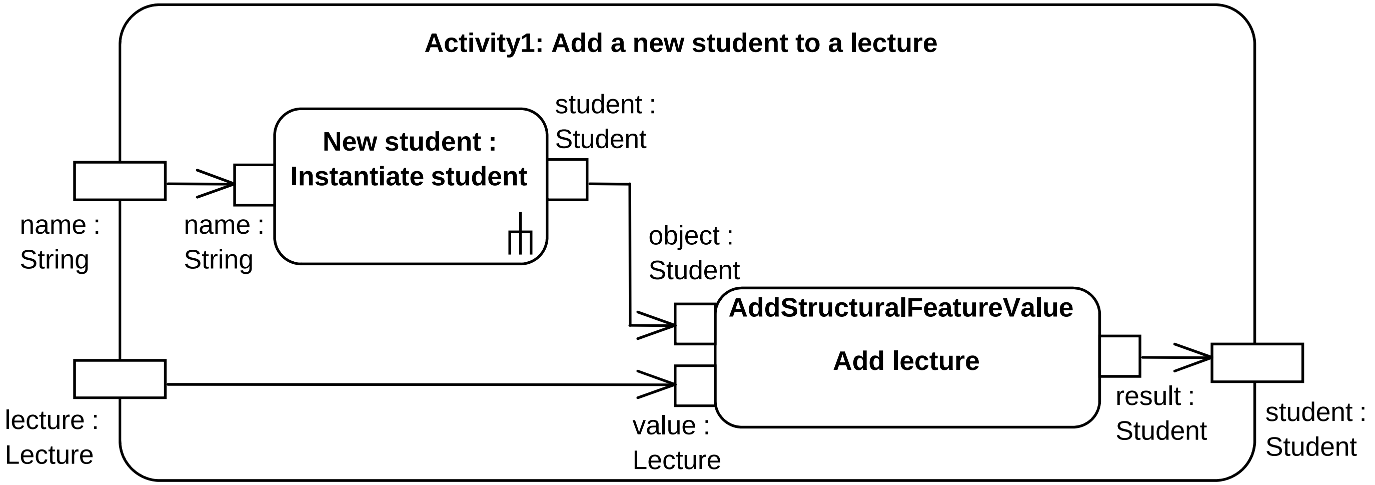
1.	**analyzing** the execution	→	**Trace model**
2.	**observing** the execution state	→	**Event model**
3.	**controlling** the execution	→	**Command API**

Contributions

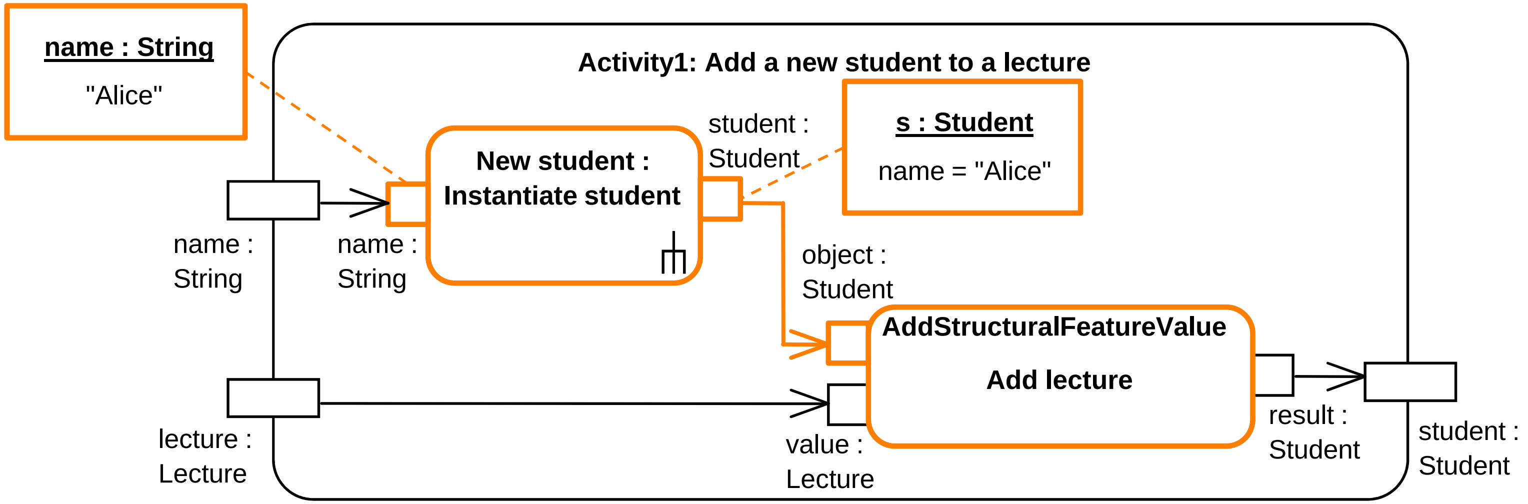
fUML virtual machine lacks in providing means for

1. **analyzing** the execution → **Trace model**
    - **Chronological execution order**
    - **Input / output relationships**
    - **Token flows**
    - **Call hierarchies**
2. **observing** the execution state → **Event model**
3. **controlling** the execution → **Command API**

Trace Model


Activity1: Add a new student to a lecture
New student : Instantiate student
student : Student
name : String
name : String
object : Student
AddStructuralFeatureValue
Add lecture
lecture : Lecture
value : Lecture
result : Student
student : Student

Input / Output Relationship



Contributions

fUML virtual machine lacks in providing means for

1. **analyzing** the execution → **Trace model**
2. **observing** the execution state → **Event model**
   - **Start / end of activity execution**
   - **Start / end of activity node execution**
   - **Suspension of activity execution**
   - **Modification of extensional values**
3. **controlling** the execution → **Command API**

Contributions

fUML virtual machine lacks in providing means for

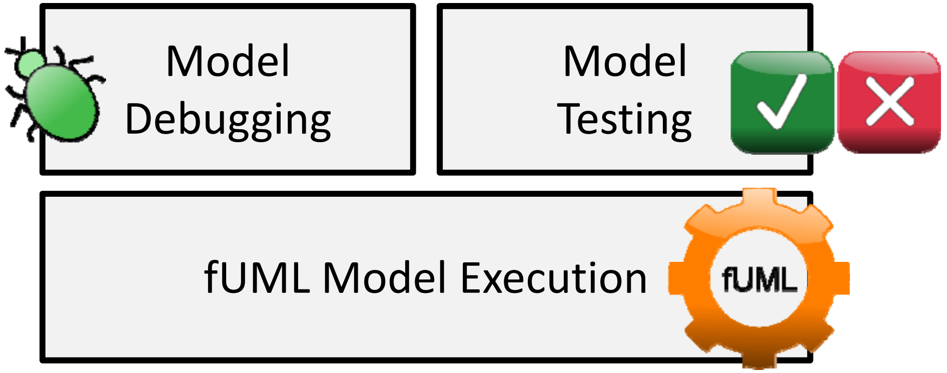
1. **analyzing** the execution → **Trace model**
2. **observing** the execution state → **Event model**
3. **controlling** the execution → **Command API**
   - **Start / stop execution**
   - **Resume execution until certain activity node**
   - **Resume execution stepwise**
   - **Retrieve runtime information**

Future Work

Objective: Model execution environment based on fUML, enabling efficient testing and debugging of models

- Impact analysis of changes
- Trace-based debugging

- DSL for specifying assertions on traces
- Automatic derivation of test cases



- Execution support for fUML → Execution support for DSML*
  - Specification of operational semantics using fUML
  - Automatic derivation of execution support

T. Mayerhofer, P. Langer, M. Wimmer. "Towards xMOF: Executable DSMLs based on fUML". Accepted for publication in: 12th Workshop on Domain-Specific Modeling at SPLASH 2012.

Thank you!

Debugging and Testing Models Based on fUML

www.modelexecution.org