Hochschule für Angewandte
Wissenschaften Hamburg
Hamburg University of Applied Sciences

Model-Based Systems Engineering
with UML/SysML and Modelica

Related Work

System Modeling, Testing and Simulation with
UML/SysML and Modelica

Parham Vasaiely, May 2010

Table of Contents

1. Introduction and Motivation
   - 1.1 Project Description
   - 1.2 Project Fields
   - 1.3 Project Progress
2. Related Work
   - 2.1 ModelicaML
   - 2.2 Papyrus
   - 2.3 A TCP/IP Based Protocol for Communicating with Real Time Simulations
   - 2.4 LiLa - Library of Labs

1. Introduction and Motivation

Problem: Increasing complexity of technical systems

- Increased degree of automation
- Increased number of involved technologies
- SW and HW development is strongly coupled
- Component suppliers are increasingly involved into the design process
- Validation of the functionality long before the first prototype is built

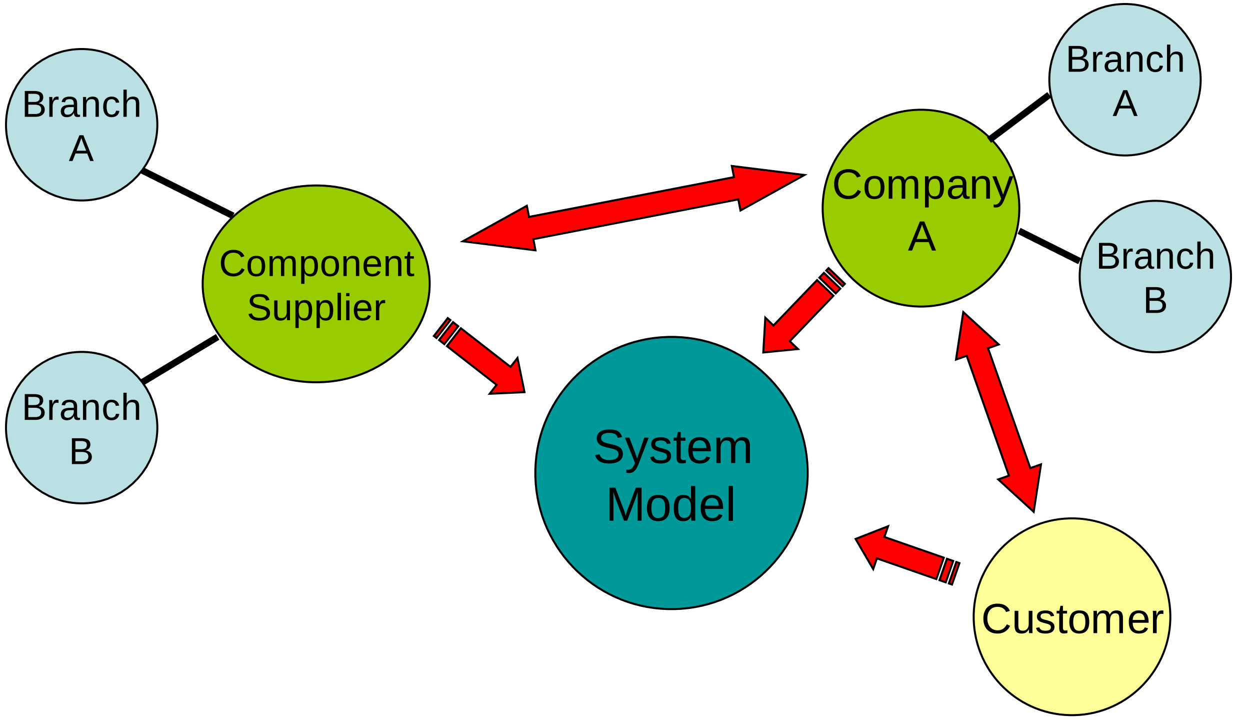
Motivation:

- Simplify and Speed systems development
- Cut time to market
- Improve Quality

Solutions:

- Model- Based Systems Engineering
- Simulation and Test

1.1 Project Description (1)



1.1 Project Description (2)

Standardized Language

Modeling based on UML
Action using Modelica

System Modeling

Environment based on Eclipse

System Simulation

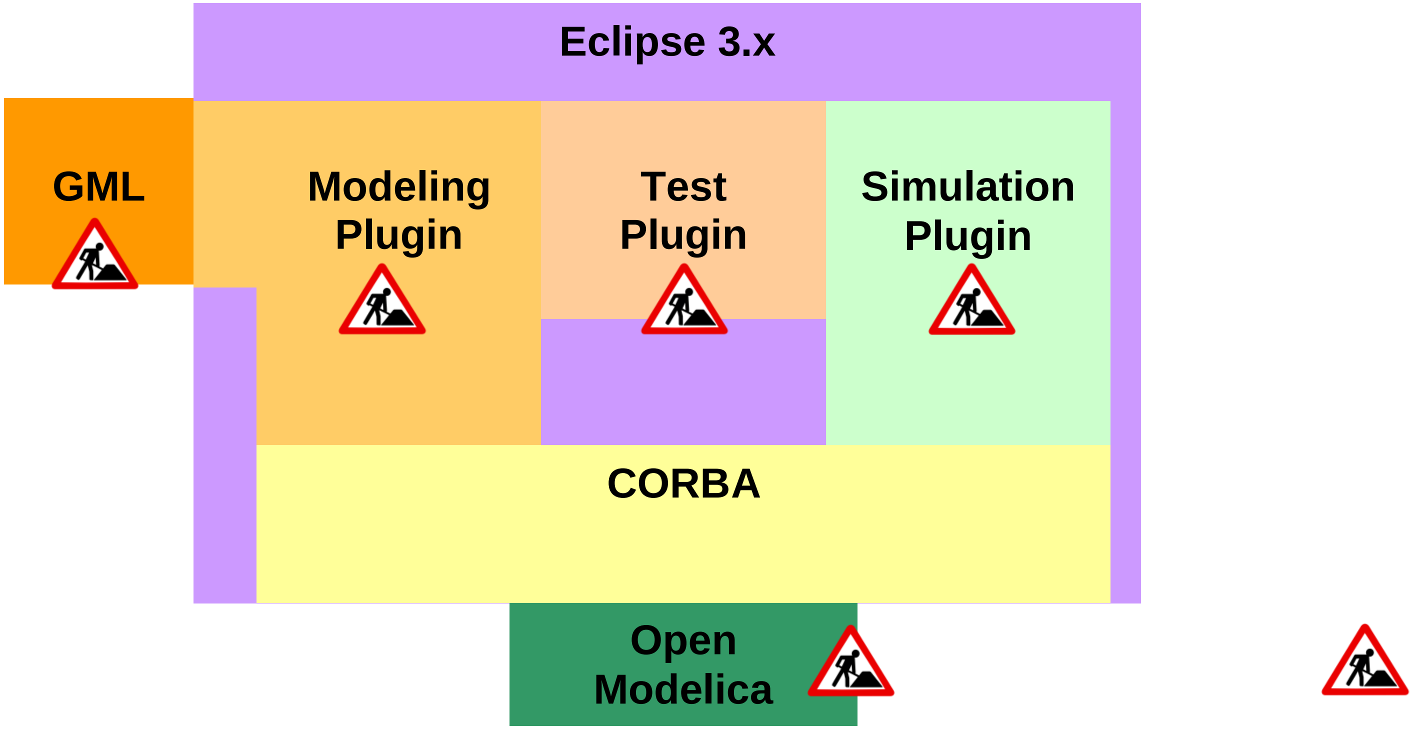
Runtime using OpenModelica
Environment based on Eclipse

Test

Environment based on Eclipse

1.2 Project Fields

Developing the following environments as Eclipse plug-ins



1.3 Project Progress

OpenModelica Simulation Runtime

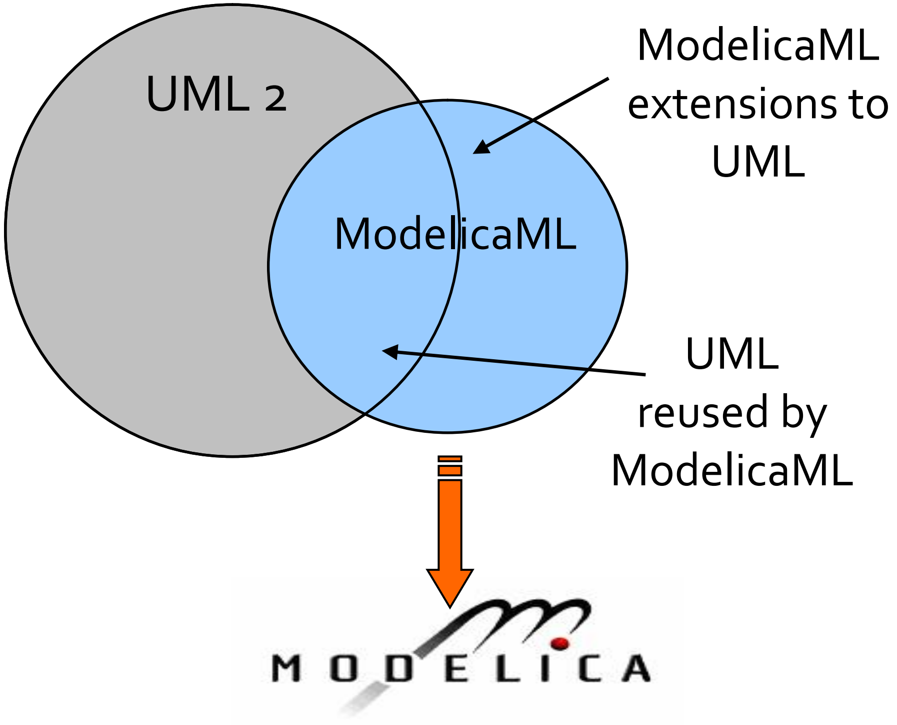
- OpenModelica Interactive

Simulation Environment

- Eclipse Plug-In Development (Prototype)

2.1 ModelicaML

Graphical Modeling Language



UML 2

ModelicaML

ModelicaML extensions to UML

UML reused by ModelicaML

MODELICA

Why ModelicaML and not pure UML?

What is wrong with SysML?

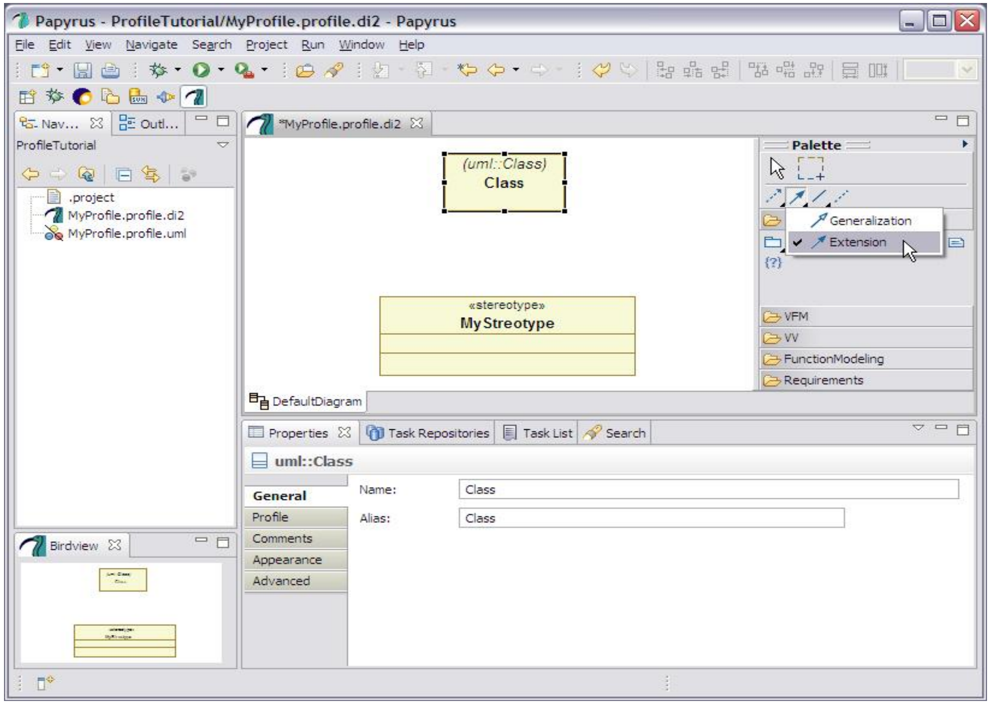
Usability and Feedback from Users…

2.2 Papyrus (1)

Eclipse Plug-In Papyrus

Create ModelicaML MetaModel using Papyrus UML Profil Editor

Papyrus
MetaModel Editor



Name: Class
Alias: Class

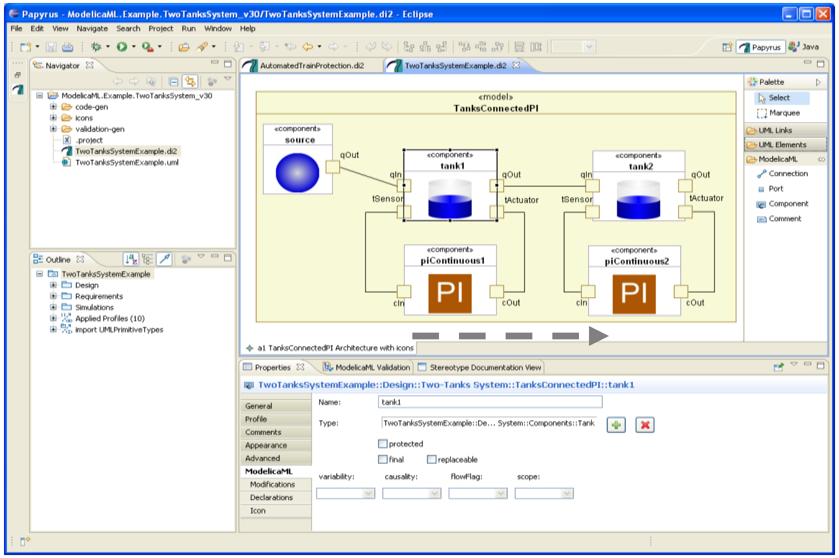
2.2 Papyrus (2)

Eclipse Plug-In Papyrus

Create Modeling Environment with Papyrus Modeling Plug-In

Papyrus UML

ModelicaML Profil


Papyrus - ModelicaML.Example.TwoTanksSystem_v30/TwoTanksSystemExample.di2 - Eclipse

TwoTanksSystemExample::Design::Two-Tanks System::TanksConnectedPI::tank1

Name: tank1

Type: TwoTanksSystemExample::De... System::Components::Tank

□ protected

□ final □ replaceable

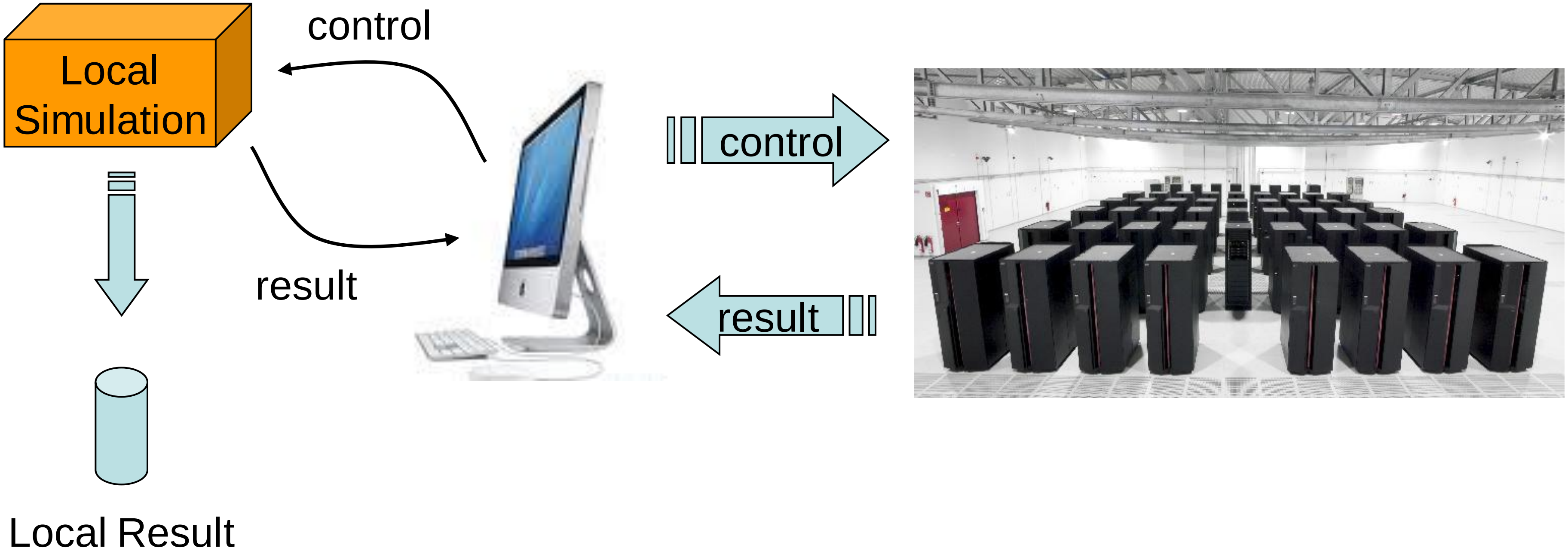
variability: causality: flowFlag: scope:

Acceleo

Model to Text Transformation

2.3 A TCP/IP Based Protocol for Communicating with Real Time Simulations (1)

Communication between Client and Simulation Runtime



Local Simulation

control

result

control

result

Local Result

2.3 A TCP/IP Based Protocol for Communicating with Real Time Simulations (2)

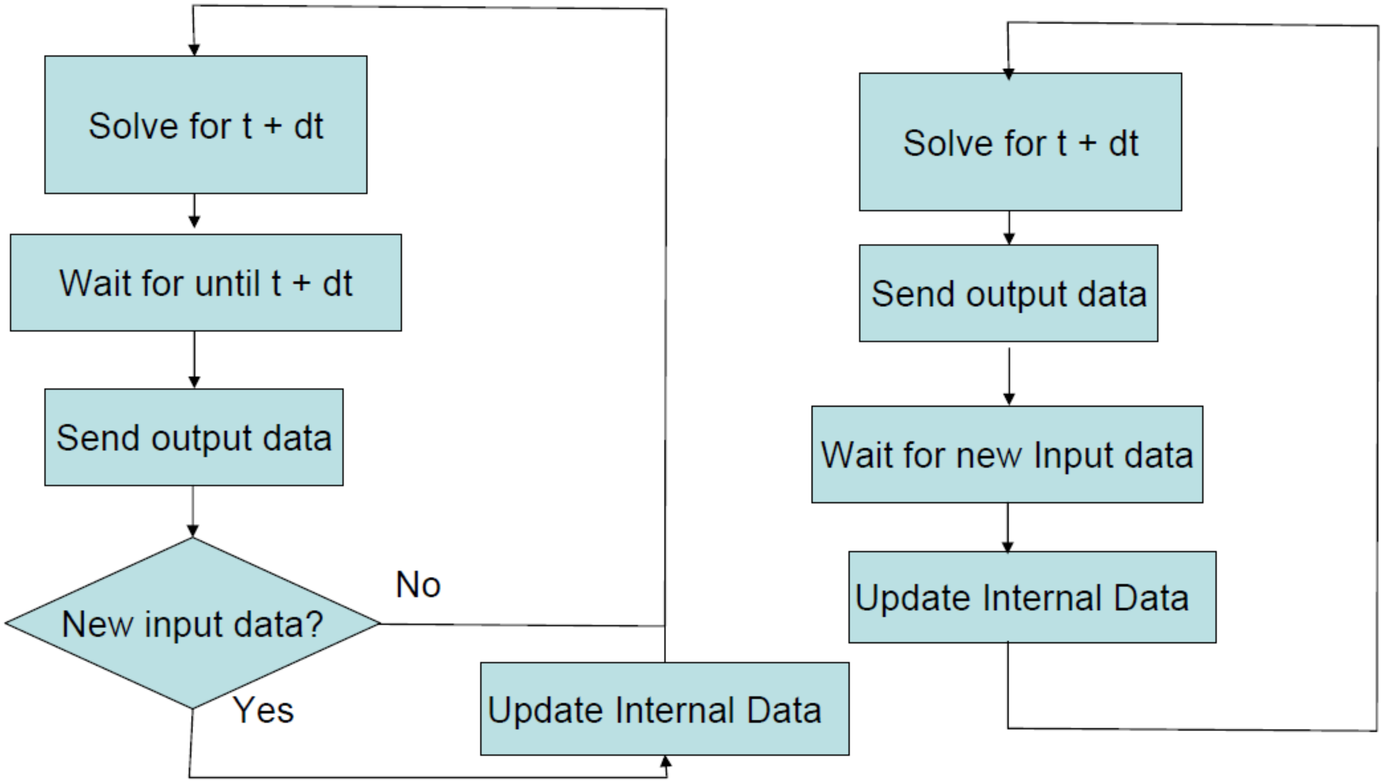
Responsible Team

MathCore Engineering AB

MathModelica

2.3 A TCP/IP Based Protocol for Communicating with Real Time Simulations (3)

Solver Loop

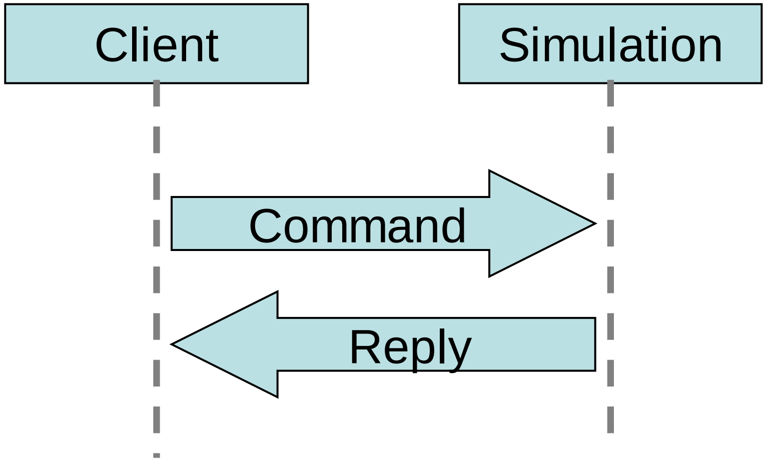


2.3 A TCP/IP Based Protocol for Communicating with Real Time Simulations (4)

Simulation Control Session

Query model information

- Model Data
- Variables
- Parameters



Simulation control

- Start, Stop, Pause
- Set input variable values
- Set parameter values
- Set initial values

Commands sent as plain text: setParameters({“p1”, 34.5, “p2”, 45.6})

2.3 A TCP/IP Based Protocol for Communicating with Real Time Simulations (5)

Result Data Transmission

Alt. 1: Send all results as plain text

result(“speedSensor.w”=0.1, …, “signalVoltage.v”=20)

Alt. 2: Send results as plain text using a Mask

result(“speedSensor.w”=0.1)

Problem of plain text used with Modelica → Full Qualified Names

“Tank1.Comp1.PartA.PartA1.PartA2….speedSensor.w”

Alt. 3: Send results using a Mask and Indexing

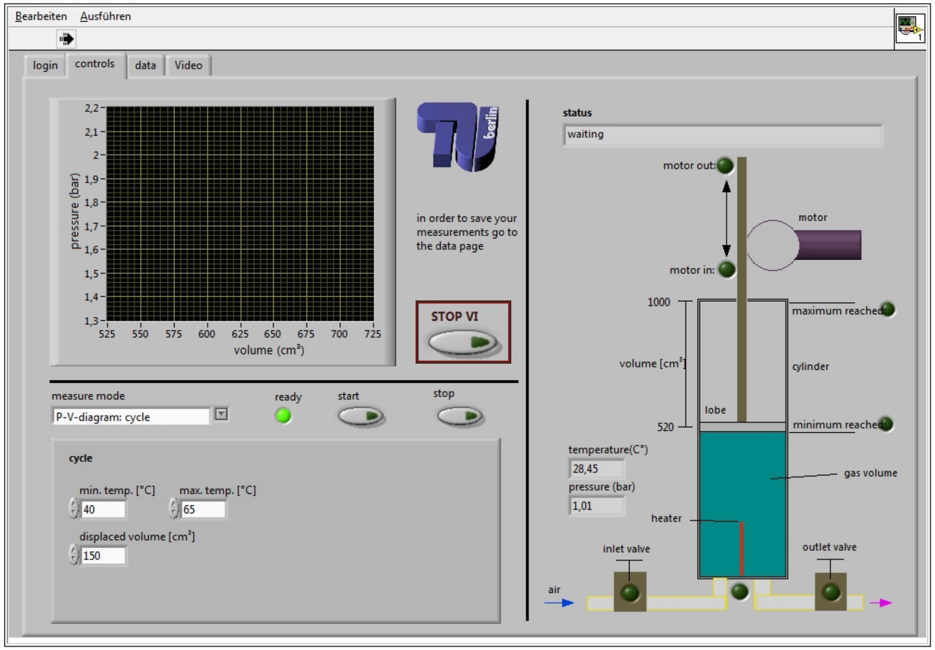
result(1=0.1, 2=20)

Thinking Forward → Use compression

2.4 LiLa – Library of Labs (1)

Virtual Control on Real Hardware

Experiments are performed on equipment that is controlled by an operator in a remote location via the internet.



in order to save your measurements go to the data page

status: waiting

measure mode: P-V-diagram: cycle

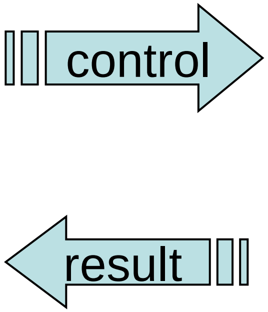
temperature(C°): 28,45

pressure (bar): 1,01

min. temp. [°C]: 40

max. temp. [°C]: 65

displaced volume [cm³]: 150

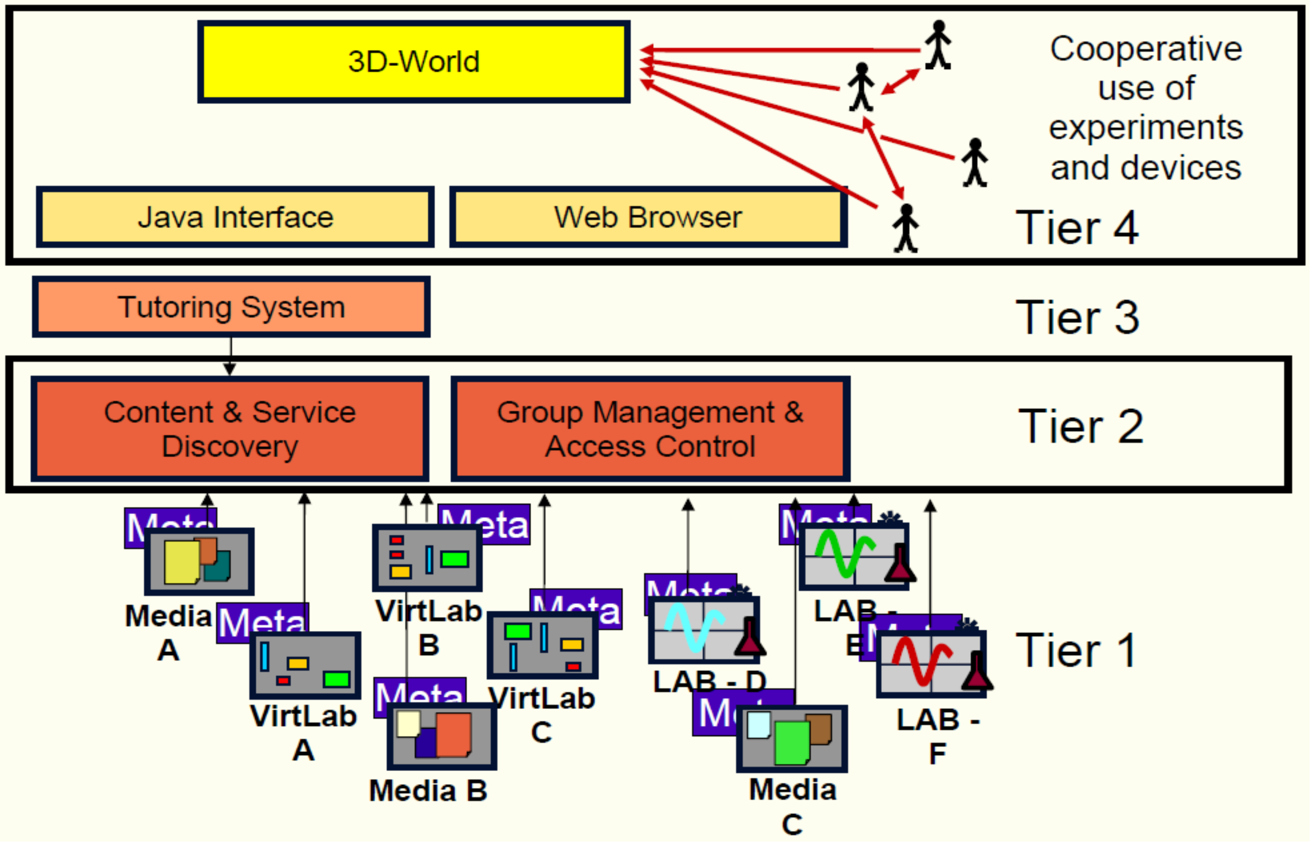


control

result

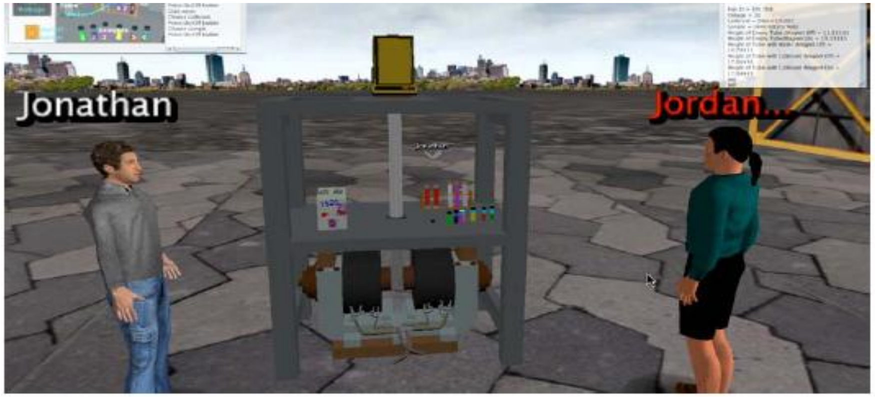
2.4 LiLa – Library of Labs (2)

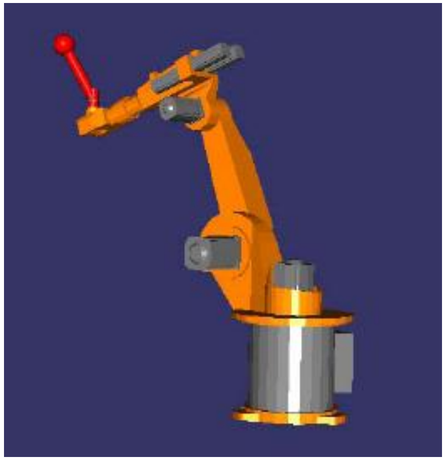
LiLa Virtual Labs



2.4 LiLa – Library of Labs (3)

LiLa Virtual Labs





Project Wonderland
File View Placemarks Shared Apps Tools Dev Tools Window Help

2.4 LiLa – Library of Labs (4)

	Manual Control Real Hardware	Virtual Control Real Hardware	LiLa Virtual Labs
Communication	Only local observ.	Over the Internet	Over the Internet
Easy Reproducable	Not Really	Not Really	Yes
Variable in Use	No	Results as Input…	Reuse Component
Observable	Limited Variables	Limited Variables	All Variables
Expensive	Yes	Yes	No
Availability	During business hours	During business hours or 24h	24h
Dangerous	Yes	Yes	No

References

- Sanford Friedenthal, Alan Moore and Rick Steiner, Practical Guide to SysML, 2008
- Fritzson Peter. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1, 2004
- Modelica Association. Modelica: A Unified Object-Oriented Language for Physical Systems Modeling: Language Specification Version 3.0, Sept 2007. www.modelica.org
- OMG. OMG Systems Modeling Language (OMG SysML™), Version 1.1, November 2008.
- Acceleo, Eclipse Plug-In. www.acceleo.org/pages/home/en
- Papyrus UML, www.papyrusuml.org
- Schamai.W, ModelicaML: a UML Profile for Modelica, http://www.ida.liu.se/~pelab/modelica/OpenModelica/MDT/ModelicaML/doc/ModelicaML_getting_started_v05_1.pdf
- OPENPROD ITEA2 Project Spec 2009 (Non public version)
- Library of Labs, http://www.lila-project.org/

Thank you for your attention!

Parham Vasaiely
Parham.Vasaiely@informatik.haw-hamburg.de