EADS INNOVATION WORKS

Systems Engineering

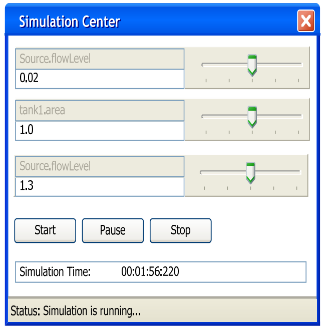
OpenModelica Interactive (OMI)

Wladimir Schamai (EADS Innovation Works, Hamburg, Germany)
Parham Vasaiely (University of Applied Science, Hamburg, Germany)

Interactive System Simulation

- Time synchronous Simulation
- Real-time Simulation
- **User Interactive Simulation**
  - **for design validation, training, failure injection, etc.**

control



Simulation Center

Source.flowLevel: 0.02

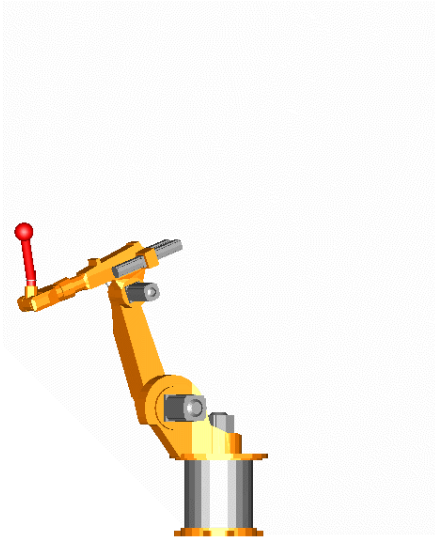
tank1.area: 1.0

Source.flowLevel: 1.3

Simulation Time: 00:01:56:220

Status: Simulation is running...

control

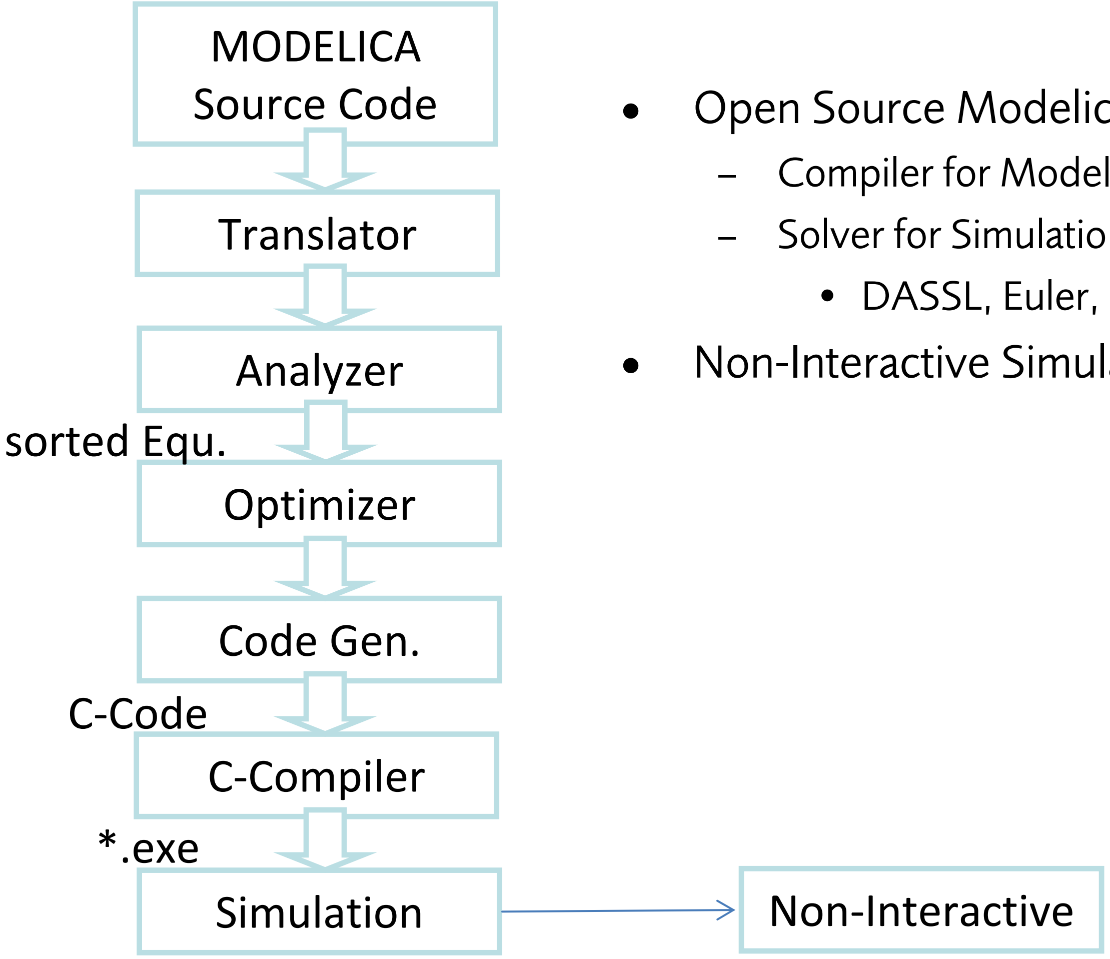


results

Model Stimulation at Run-Time

- Input variables can be modified at run time at the top level of the model
  - No change of the Modelica semantics
- Parameters can be modified (using the full-qualified name) at run time at any level of the model hierarchy
  - This is a relaxation/enhancement of Modelica semantics
  - This enables a flexible way to provide user interaction without enforcing the explicit definition of top-level input variables

OpenModelica



MODELICA Source Code

Translator

Analyzer

sorted Equ.

Optimizer

Code Gen.

C-Code

C-Compiler

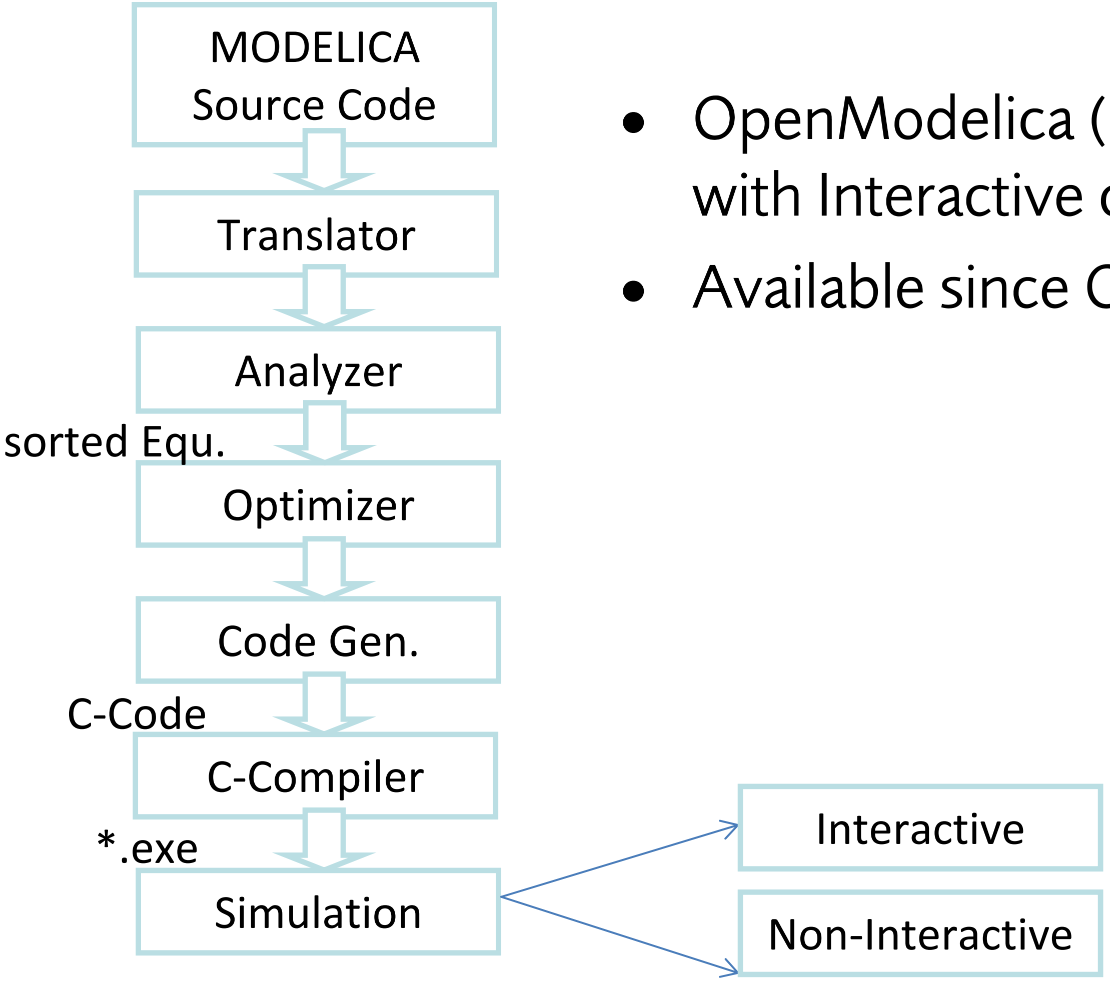
*.exe

Simulation

Non-Interactive

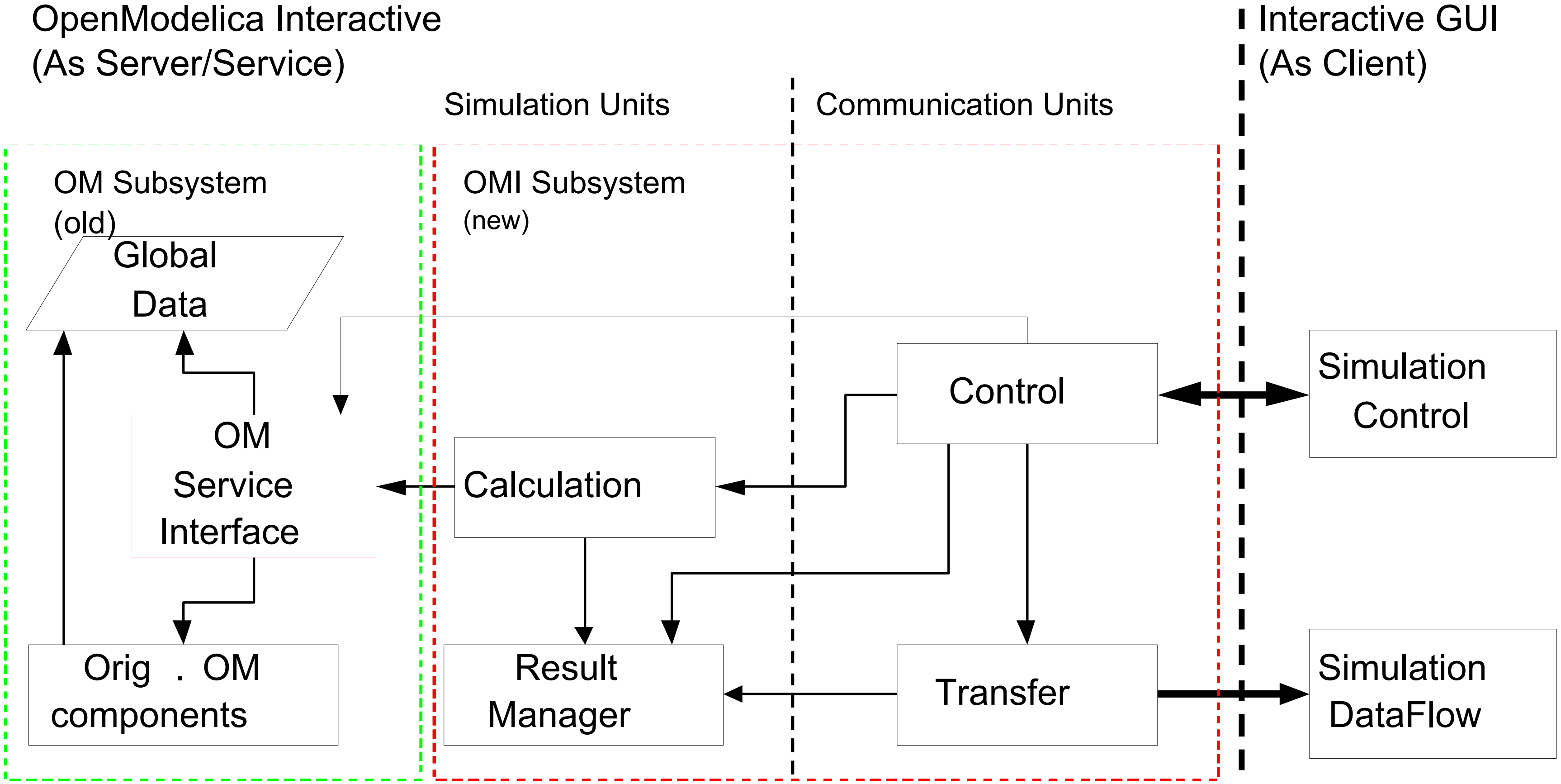
- Open Source Modelica Tool
  - Compiler for Modelica Code
  - Solver for Simulation
    - DASSL, Euler, Rungekutter
- Non-Interactive Simulation

OpenModelica Interactive (OMI)



- OpenModelica (OM) with Interactive capabilities
- Available since OM version 1.6.0

OMI System Architecture Overview



OMI Subsystem Components

- **OMI::Control**

  The "Control" module is the interface between OMI and the clients, e.g. control GUIs. It receives commands (messages) from clients and controls the simulation accordantly. It communicates the simulation status to clients.

- **OMI::ResultManager**

  The "ResultManager" is responsible for organizing simulation result data, provides synchronized access (between calculation and transfer) to simulation result data.

- **OMI::Calculation**

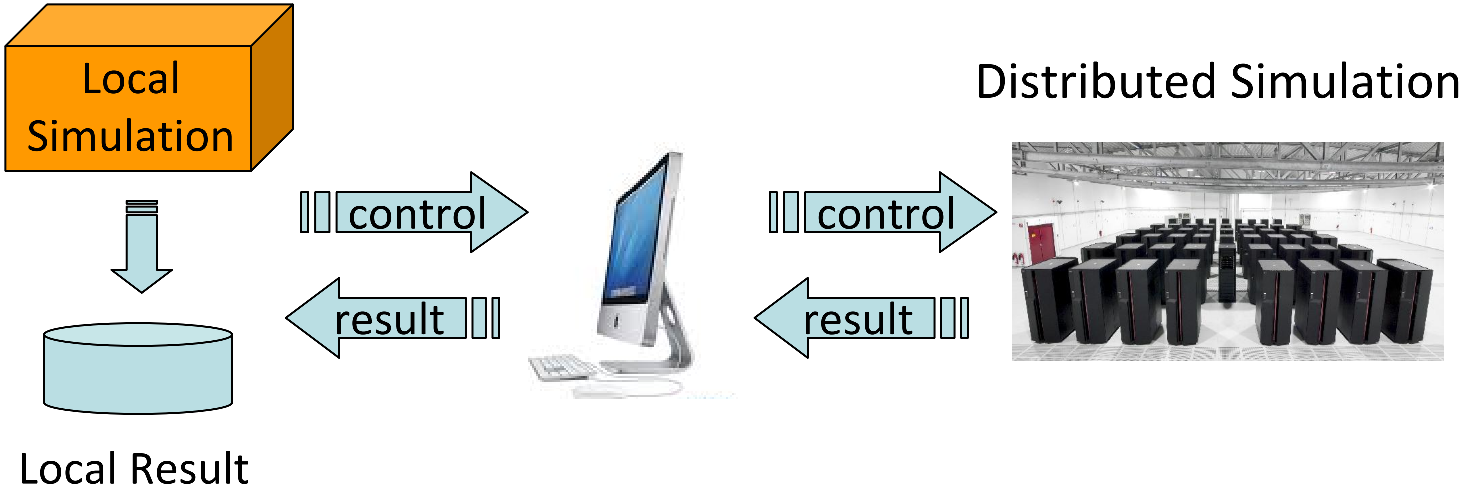
  The "Calculation" thread uses the "OM Solving Service" to get results for a specific time step and to inform the "ResultManager" about the new simulation results. Before each time step it allows to modify the model, i.e. it forwards user stimuli to the running Modelica model.

- **OMI::Transfer**

  "Transfer" thread gets simulation results from the "ResultManager", synchronizes with the real time and forward the simulation results to clients, e.g. GUIs.

Communication Technology

- CORBA approach is overload for this purpose
- Message passing, based on TCP/IPv4, is used as communication technology, which also allows the usage of parallel computation



Communication Components

OMI server and client components

Name	Description	URL
Control Server	Waits for requests from the GUI	Waits for connection on: **127.0.0.1:10501**
Control Client	Replies to the GUI and sends other synchronization messages to it	Tries to connect on: 127.0.0.1:10500
Transfer Client	Sends simulation results to a GUI	Tries to connect on: 127.0.0.1:10502

GUI server and client components

Name	Description	URL
Control Client	Requests to the OMI Control Server	Tries to connect on: **127.0.0.1:10501**
Control Server	Waits for information from the OMI Control Client	Waits for connection on: 127.0.0.1:10500
Transfer Server	Waits for simulation results from the OMI Transfer Client	Waits for connection on: 127.0.0.1:10502

Operation Messages

Message Passing using String Message

prefix: Operation e.g.: start, stop, etc.

sequence: A counter used to verify the execution of an operation

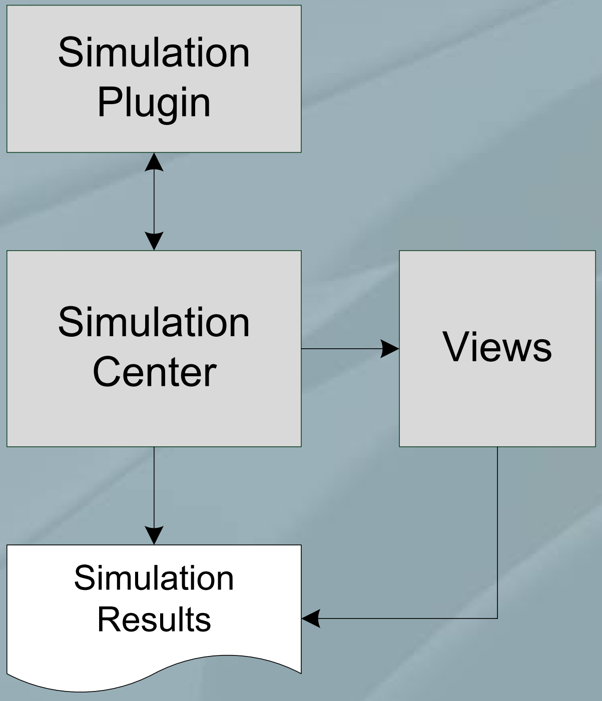
message: parameter e.g. Parameter name

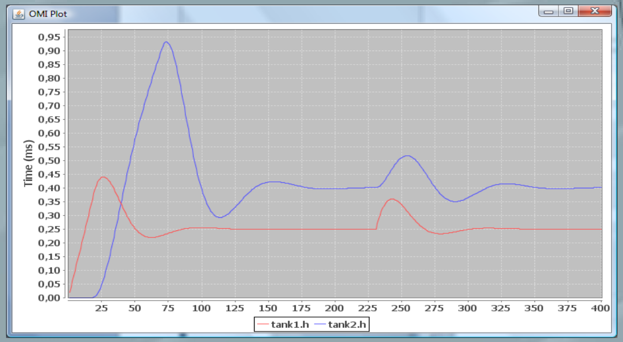
suffix: end

GUI Request	Description	OMI::Control Reply
start#1#end	Starts or continues the simulation	done#1#end
stop#3#end	Stops the running simulation and resets all values to the beginning	done#3#end
changevalue#5#100#par1=2.3:par2=33.3#end	Changes the value of the appended parameters (par1=2.3, par2=33.3) at time 100 and stets the simulation time back to the point where the user clicked in the GUI (i.e. results after this point in time are deleted in order to ensure the synchronization with real interaction time)	done#5#end

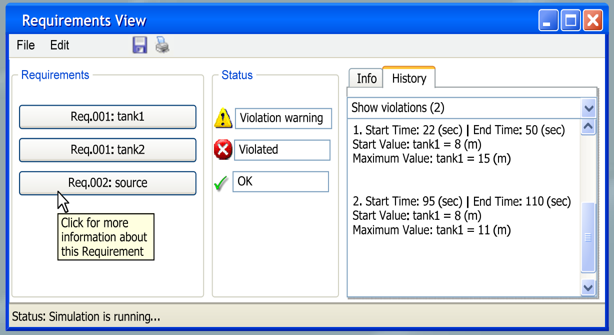
Simulation Visualization Examples

Example of a Eclipse based Visualization Tool





Plot View



Requirements

Req.001: tank1

Req.001: tank2

Req.002: source

Click for more information about this Requirement

Status

⚠ Violation warning

☒ Violated

✓ OK

Info | History

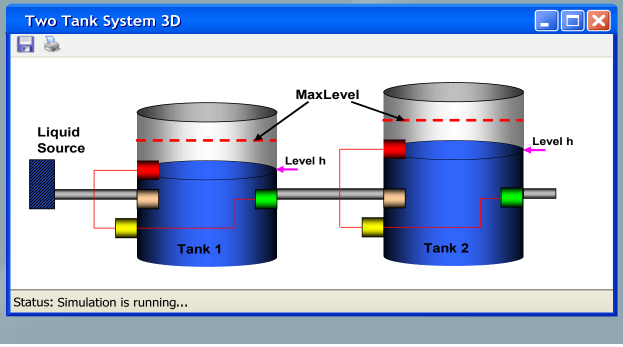
Show violations (2)

1. Start Time: 22 (sec) | End Time: 50 (sec)
Start Value: tank1 = 8 (m)
Maximum Value: tank1 = 15 (m)

2. Start Time: 95 (sec) | End Time: 110 (sec)
Start Value: tank1 = 8 (m)
Maximum Value: tank1 = 11 (m)

Status: Simulation is running...

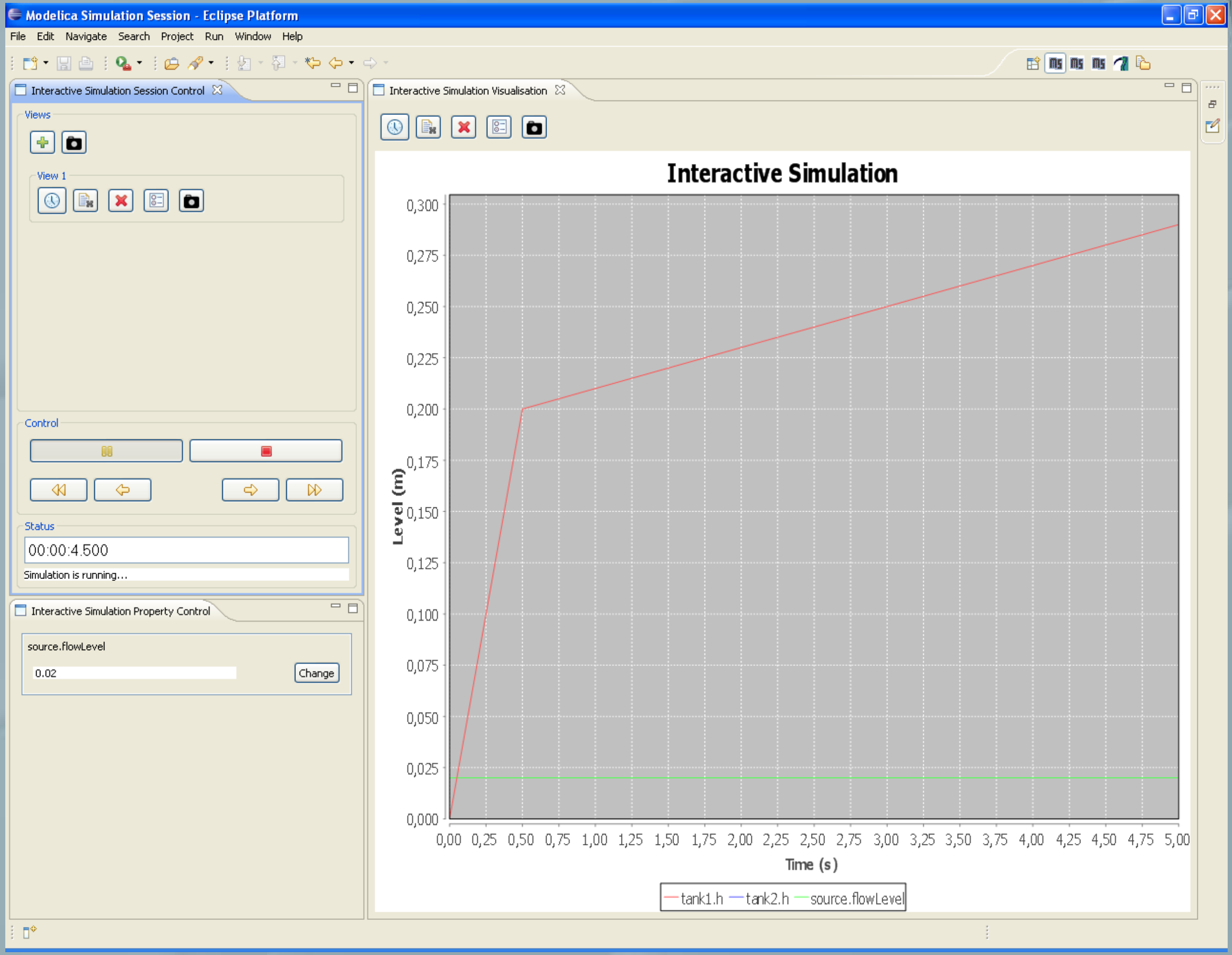
Requirements Evaluation View



Status: Simulation is running...

Domain-Specific Visualization View

Live-Demo



Modelica Simulation Session - Eclipse Platform

Interactive Simulation Session Control

Status

00:00:4.500

Simulation is running...

Interactive Simulation Property Control

source.flowLevel

0.02

Outlook

- OMC enhancements:
  - Enable dassl2 to be used for interactive simulations
  - Improve error handling

- Implementation of Eclipse plug-in "Interactive Simulation Center"

- Implementation of example of standalone control GUIs and domain specific visualizations