

OpenModelica SimulationRuntime Interface - OMSI?

Status and Plans

Willi Braun¹ Niklas Worschec² Bernhard Bachmann¹

FH Bielefeld University of Applied Sciences¹
Bielefeld, Germany

Bosch Rexroth²
Lohr, Germany



February 19, 2018

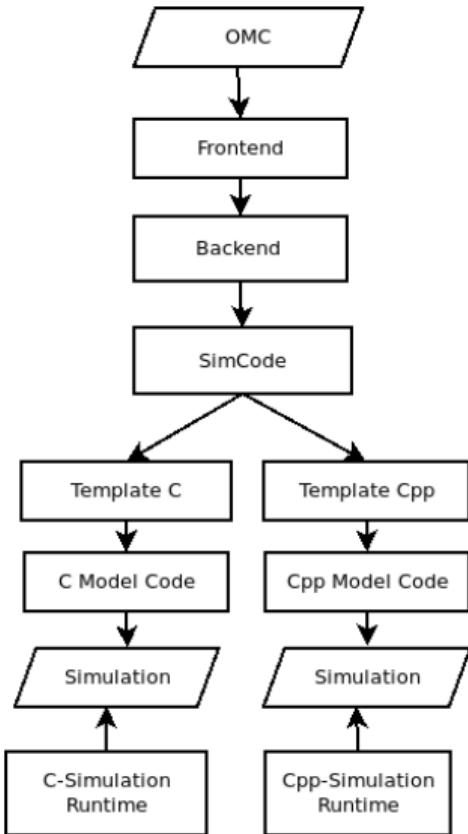
SPONSORED BY THE



Federal Ministry
of Education
and Research

Motivation

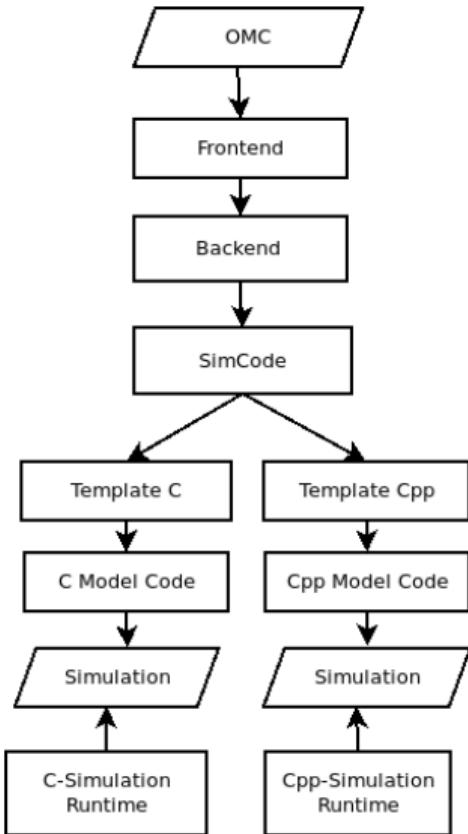
Current issues



- Every development in the Backend require independent development in the Templates
- Differences in Backend and SimCode
- The Templates contain a lot of similar code
- No clear separation between model data and runtime data

Motivation

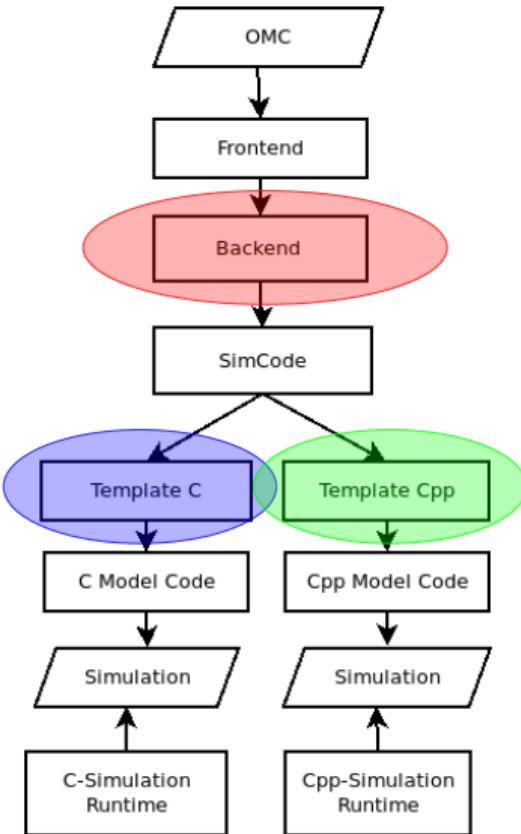
Current issues



- Every development in the Backend require independent development in the Templates
- Differences in Backend and SimCode
- The Templates contain a lot of similar code
- No clear separation between model data and runtime data

Motivation

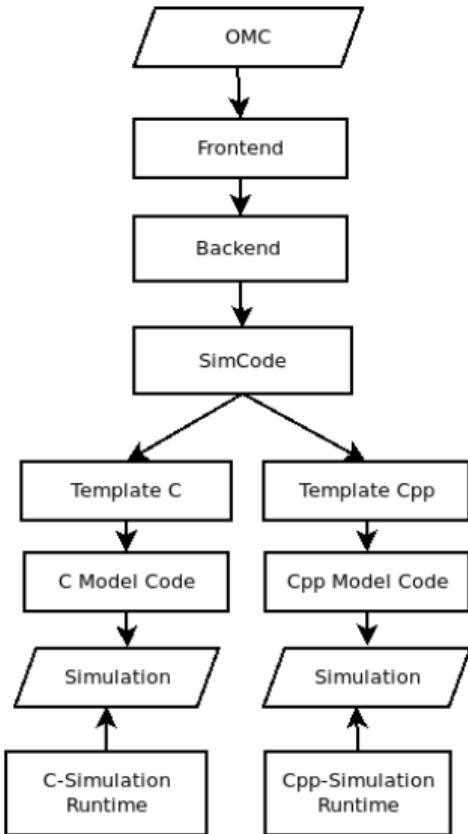
Current issues



- Every development in the Backend require independent development in the Templates
- Differences in Backend and SimCode
- The Templates contain a lot of similar code
- No clear separation between model data and runtime data

Motivation

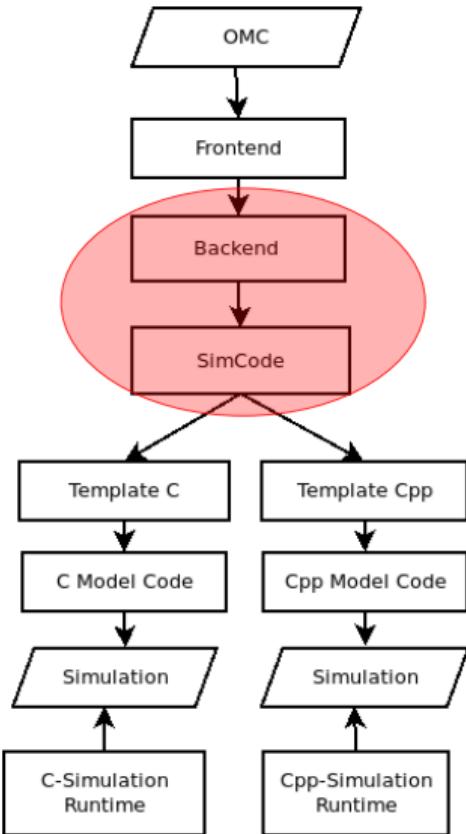
Current issues



- Every development in the Backend require independent development in the Templates
- Differences in Backend and SimCode
- The Templates contain a lot of similar code
- No clear separation between model data and runtime data

Motivation

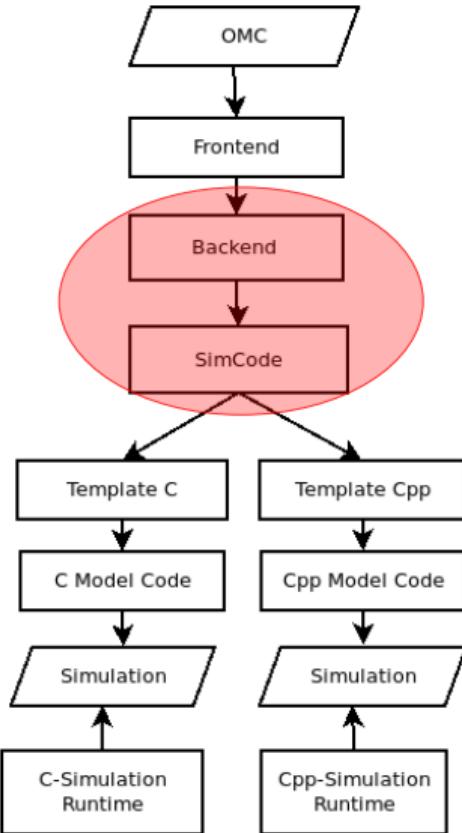
Current issues



- Every development in the Backend require independent development in the Templates
- Differences in Backend and SimCode
- The Templates contain a lot of similar code
- No clear separation between model data and runtime data

Motivation

Current issues

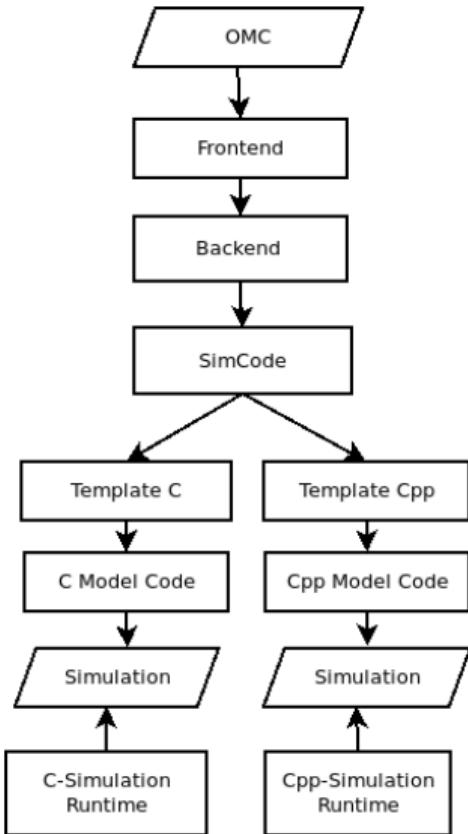


- Every development in the Backend require independent development in the Templates
- Differences in Backend and SimCode

```
wbraun@trestle:~/workspace/OpenModelica/OMCompiler/Compiler$ git grep 'ifcpp|\"Cpp\"' BackEnd/ SimCode/  
BackEnd/BackendDAECreate.mo:    true = stringEq(Flags.getConfigString(Flags.SIMCODE_TARGET), "Cpp");  
BackEnd/BackendDump.mo: if stringEqual(Config.simCodeTarget(), "Cpp") then  
BackEnd/ExpressionSolve.mo: if not stringEqual(Config.simCodeTarget(), "Cpp") then  
BackEnd/HpcMemory.mo: if(stringEqual(Config.simCodeTarget(), "Cpp")) then  
BackEnd/Initialization.mo: if stringEq(Config.simCodeTarget(), "Cpp") then  
BackEnd/Tearing.mo: //false = stringEqual(Config.simCodeTarget(), "Cpp");  
SimCode/SimCodeMain.mo: case "Cpp"  
SimCode/SimCodeMain.mo: case (,"Cpp")  
SimCode/SimCodeUtil.mo: Boolean ifcpp;  
SimCode/SimCodeUtil.mo: ifcpp := stringEqual(Config.simCodeTarget(), "Cpp");  
SimCode/SimCodeUtil.mo: if ifcpp then  
SimCode/SimCodeUtil.mo: if(boolAnd(ifcpp,Flags.getConfigBool(Flags.LABELED_REDUCTION))) then  
SimCode/SimCodeUtil.mo: if(ifcpp) then  
SimCode/SimCodeUtil.mo: if(ifcpp) then  
SimCode/SimCodeUtil.mo: if (Config.simCodeTarget() == "Cpp") then  
SimCode/SimCodeUtil.mo: if stringEqual(Config.simCodeTarget(), "Cpp") then  
SimCode/SimCodeUtil.mo: Boolean isCpp = Config.simCodeTarget() == "Cpp";  
SimCode/SimCodeUtil.mo: stateVar := listGet(inStateVars, inIndex + 1 - (if Config.simCodeTarget() == "Cpp" then 0 else listLength(inStateVars)) /* SimVar indexes start from zero */);  
SimCode/SimCodeUtil.mo: case ( , , "Cpp") algorithm
```

Motivation

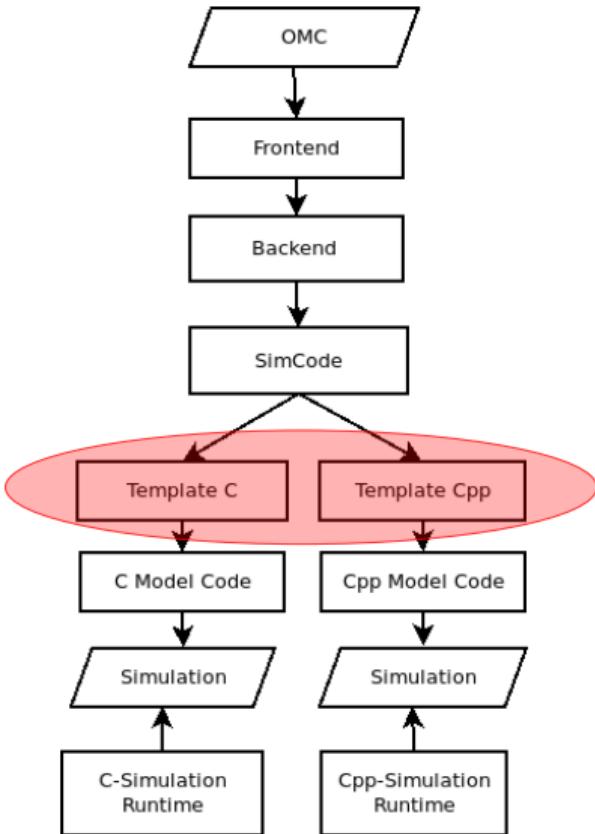
Current issues



- Every development in the Backend require independent development in the Templates
- Differences in Backend and SimCode
- The Templates contain a lot of similar code
- No clear separation between model data and runtime data

Motivation

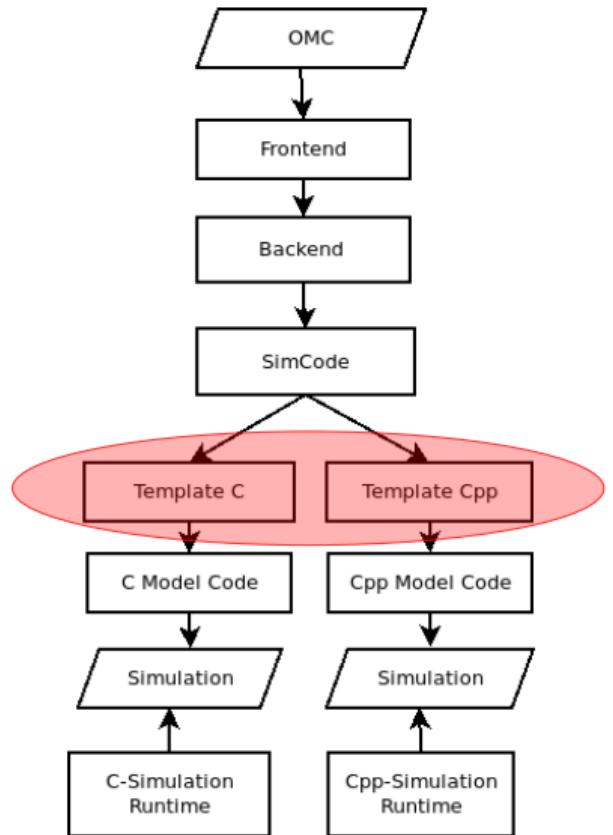
Current issues



- Every development in the Backend require independent development in the Templates
- Differences in Backend and SimCode
- **The Templates contain a lot of similar code**
- No clear separation between model data and runtime data

Motivation

Current issues



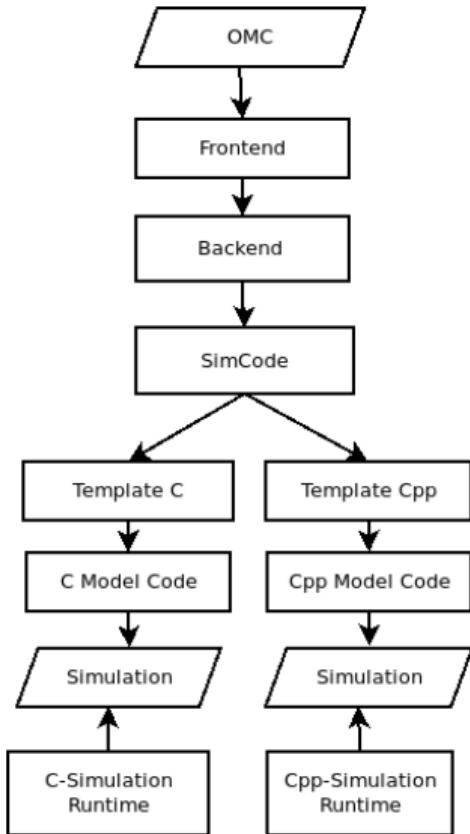
```
4647 {  
4648     TRACE_PUSH  
4649     TRACE_POP  
4650     return 1;  
4651 }  
4652 >>  
4653 case _ then  
4654     let _eachRefParts = buffer ""  
4655     let sp_size_index = listLengthElements(unzipSecond(sparsespattern))  
4656     let sizeIndex = (length(sparsespattern))  
4657     let colPtr = genSPCNSPtr((listLength(sparsespattern)), sparsespattern, "colptrIndex")  
4658     let rowIndex = genSPCRSProws((listLength(unzipSecond(sparsespattern))), sparsespattern)  
4659     let colorString = genSPColors(colorList, "data->simulationInfo->analyticJacobians")  
4660     let indexColumn = (jacobianColumn |> JAC_COLUMN((numberOfResultVars => (<\n>`  
4661     let tmpvarsSize = (jacobianColumn |> JAC_COLUMN(columnVars=vars) => listLength(var  
4662     let index = listLength(seedVars)  
4663     <\n>`  
4664     OMC_DISABLE_OPT  
4665     int <\n>`  
4666     <\n>`  
4667     TRACE_PUSH  
4668     DATA* data = ((DATA*)inData);  
4669     int index = <\n>`  
4670     <\n>`  
4671     <\n>`  
4672     int i = 0;  
4673     <\n>`  
4674     data->simulationInfo->analyticJacobians[index].sizeCols = <\n>`  
4675     data->simulationInfo->analyticJacobians[index].sizeRows = <\n>`  
4676     data->simulationInfo->analyticJacobians[index].sizeTmpVars = <\n>`  
4677     data->simulationInfo->analyticJacobians[index].seedVars = (modelica real)* callo  
4678     data->simulationInfo->analyticJacobians[index].resultVars = (modelica real)* cal  
4679     data->simulationInfo->analyticJacobians[index].tmpVars = (modelica real)* callc  
4680     data->simulationInfo->analyticJacobians[index].sparsePattern.leadIndex = (unsigned  
4681     data->simulationInfo->analyticJacobians[index].sparsePattern.index = (unsigned int  
4682     data->simulationInfo->analyticJacobians[index].sparsePattern.numberOfZeroes =  
4683     data->simulationInfo->analyticJacobians[index].sparsePattern.colorCols = (unsigned  
4684     data->simulationInfo->analyticJacobians[index].sparsePattern.maxColors = <\n>`  
4685     data->simulationInfo->analyticJacobians[index].jacobina = NULL;  
4686     /* write lead index of compressed sparse column */  
4687     memcpy(data->simulationInfo->analyticJacobians[index].sparsePattern.leadIndex, c  
4688     for(i=2;i<=sizeIndex;i+=1);++i)  
4689     data->simulationInfo->analyticJacobians[index].sparsePattern.leadIndex[i] = d  
4690     <\n>`  
4691 }
```

- The Templates contain a lot of similar code

- No clear separation between model data and runtime data

Motivation

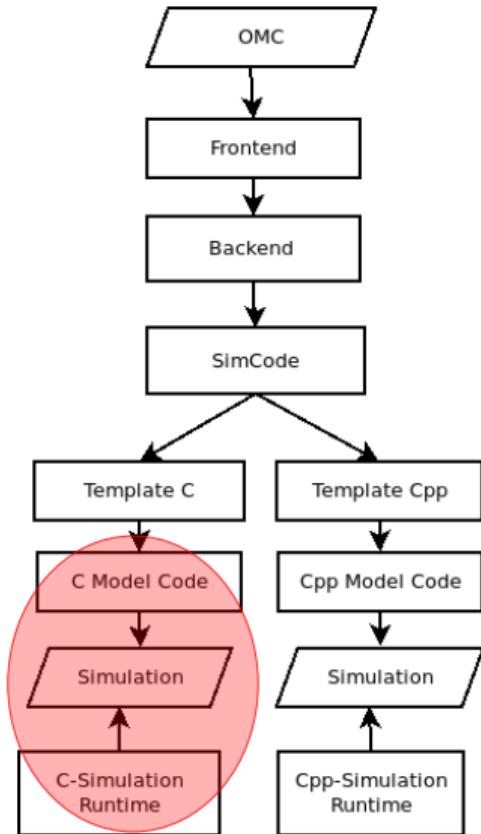
Current issues



- Every development in the Backend require independent development in the Templates
- Differences in Backend and SimCode
- The Templates contain a lot of similar code
- No clear separation between model data and runtime data

Motivation

Current issues



- Every development in the Backend require independent development in the Templates
- Differences in Backend and SimCode
- The Templates contain a lot of similar code
- **No clear separation between model data and runtime data**

Goal: Specify a clear simulation model interface with a uniform data model.

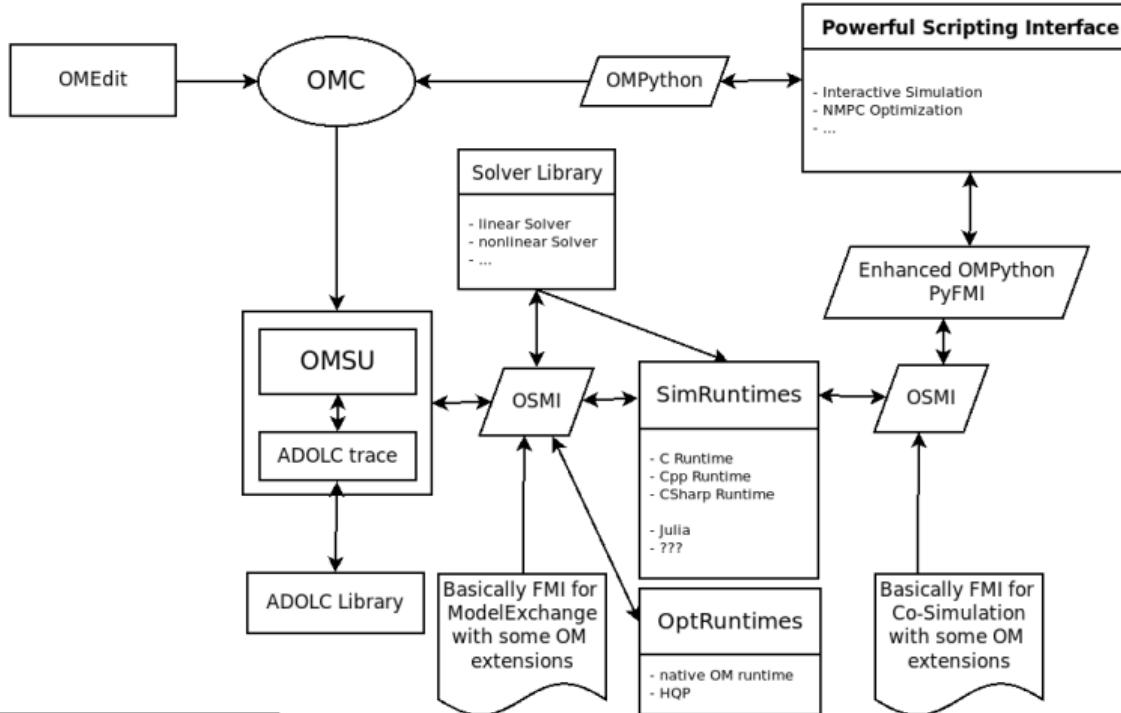
- Outline:
 - ▶ Overview
 - ▶ Current Concept
 - ▶ Status & Outlook

Goal: Specify a clear simulation model interface with a uniform data model.

- Outline:
 - ▶ Overview
 - ▶ Current Concept
 - ▶ Status & Outlook

Current Concept

Overview¹



¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModellInterface ? (see slide 20)

Current Concept

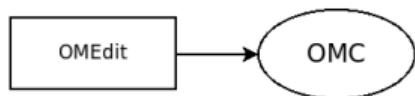
Overview¹



¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModellInterface ? (see slide 20)

Current Concept

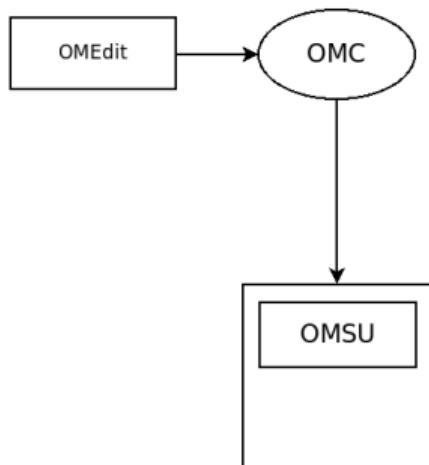
Overview¹



¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModellInterface ? (see slide 20)

Current Concept

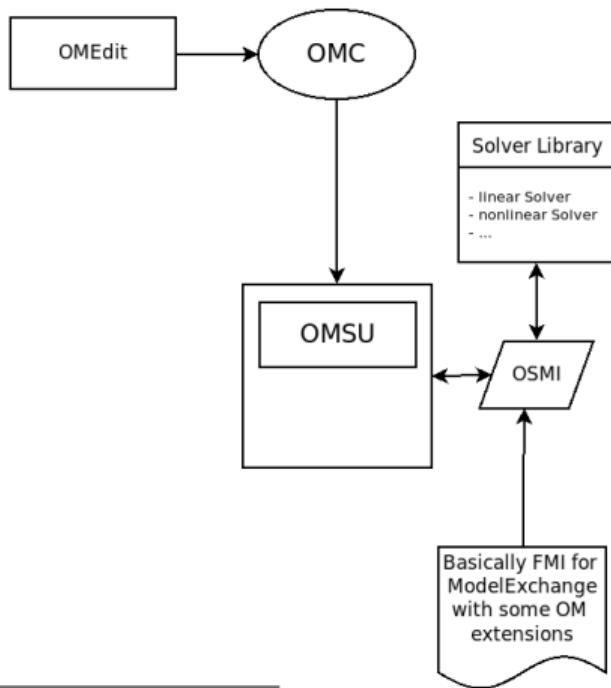
Overview¹



¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModellInterface ? (see slide 20)

Current Concept

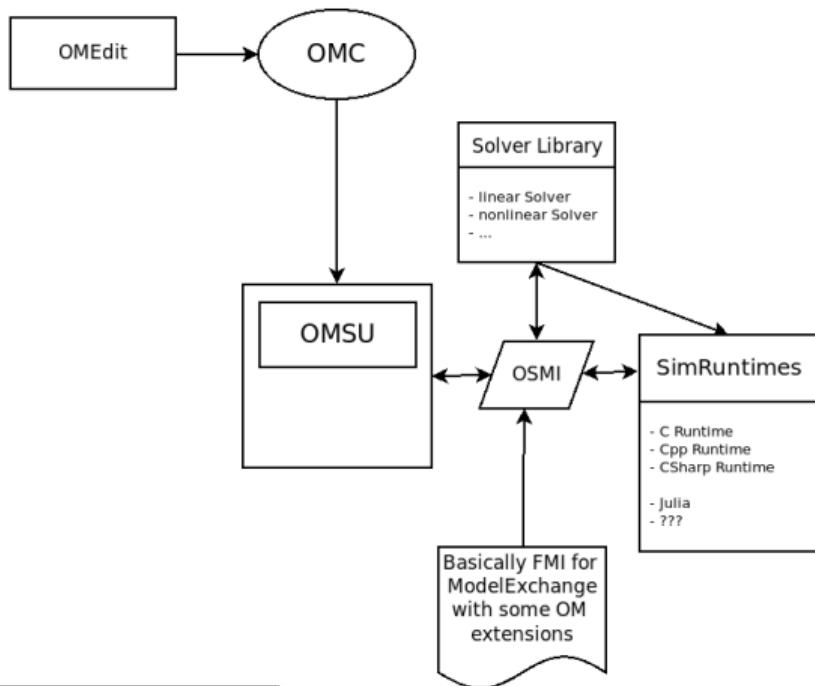
Overview¹



¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModelInterface ? (see slide 20)

Current Concept

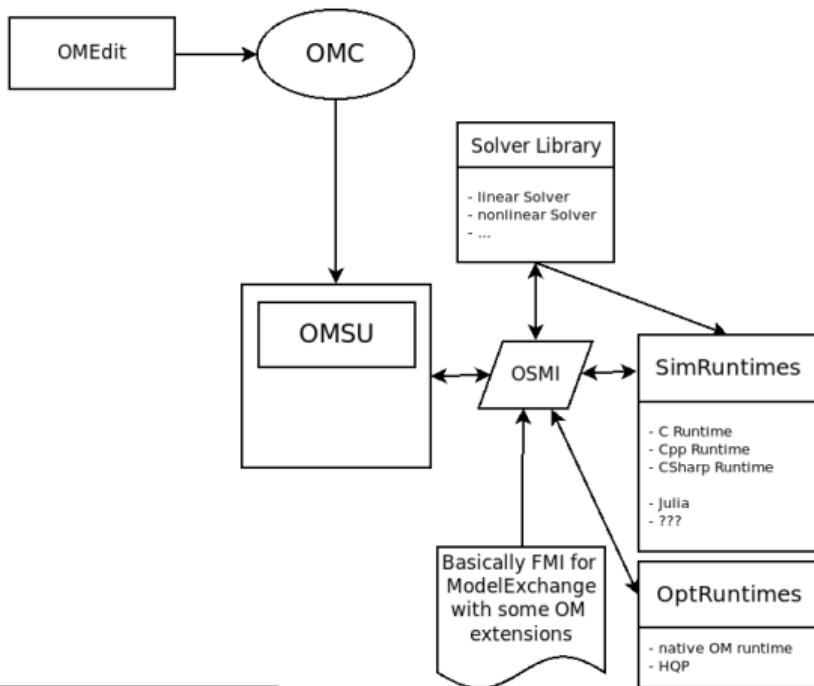
Overview¹



¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModelInterface ? (see slide 20)

Current Concept

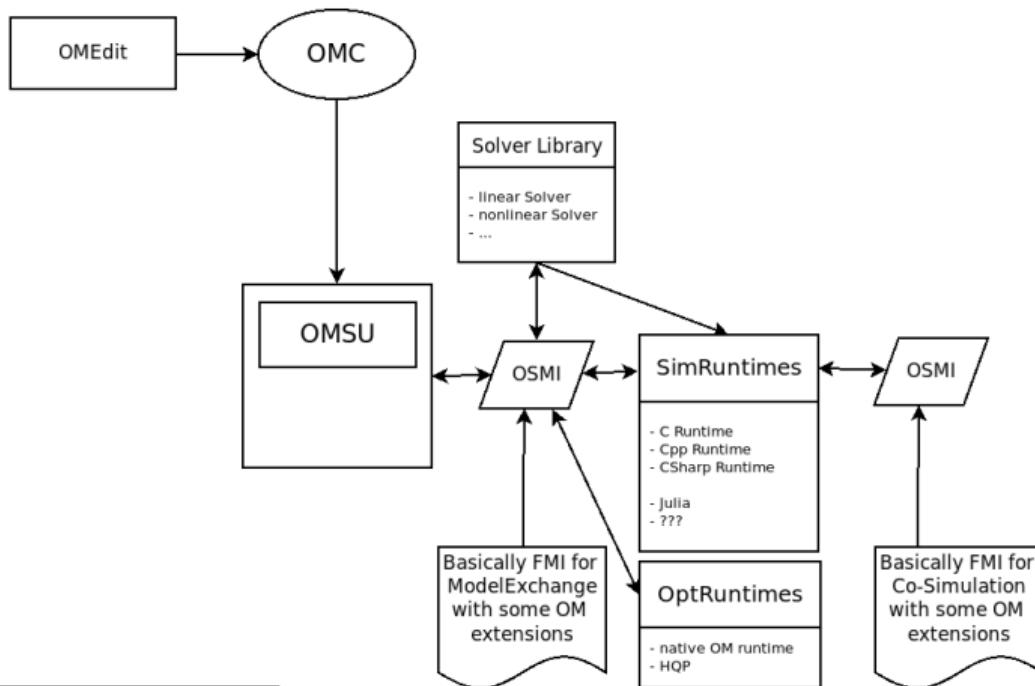
Overview¹



¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModellInterface ? (see slide 20)

Current Concept

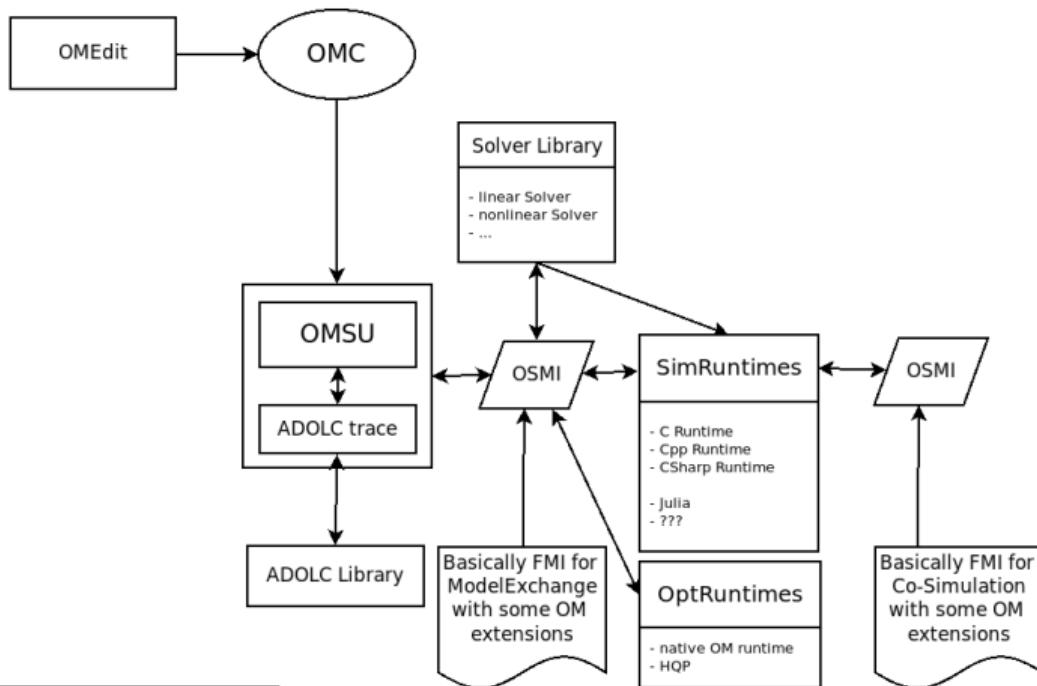
Overview¹



¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModellInterface ? (see slide 20)

Current Concept

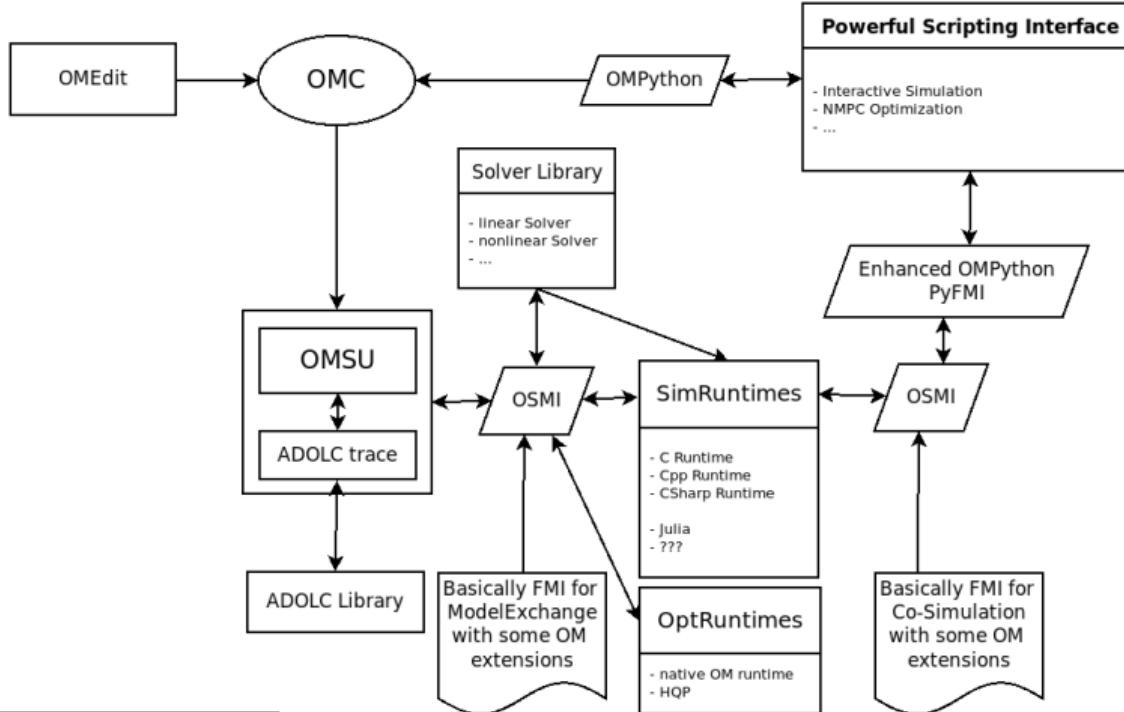
Overview¹



¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModellInterface ? (see slide 20)

Current Concept

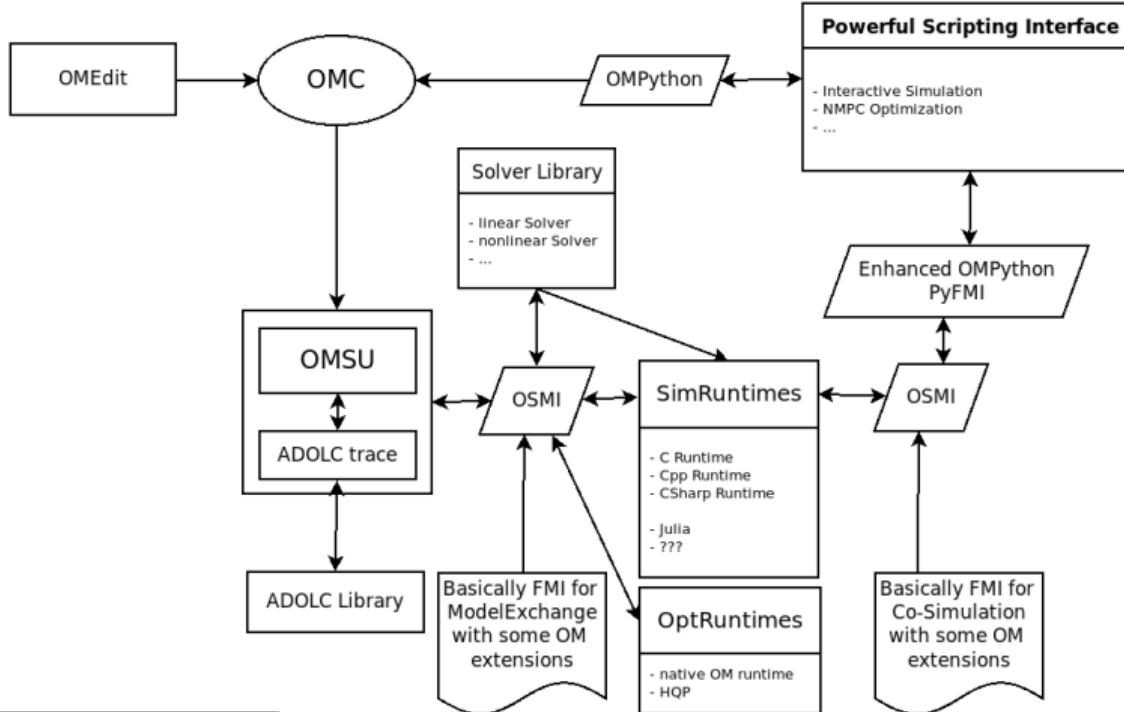
Overview¹



¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModellInterface ? (see slide 20)

Current Concept

Overview¹



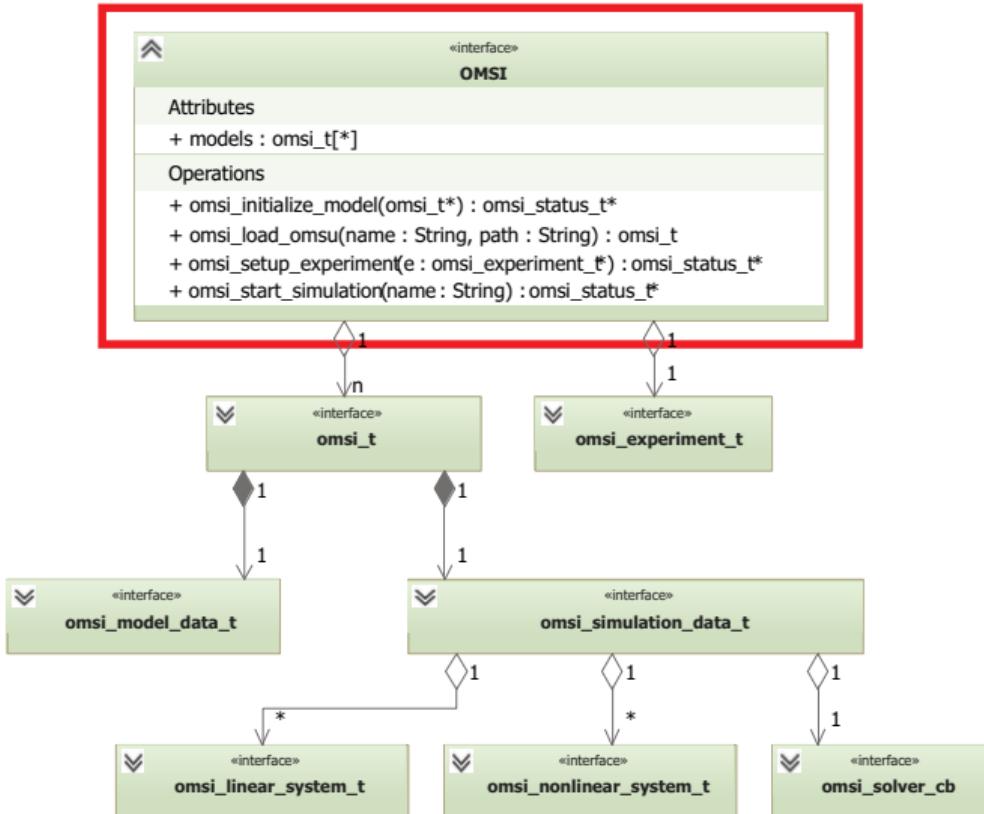
¹ OMSI -> OpenModelicaSimulationInterface or OSMI -> OpenmodelicaSimulationModellInterface ? (see slide 20)

- OpenModelica Simulation Interface (OMSI)

- ▶ Unification of the C and C++ Runtime
- ▶ Simulation interfaces include functions for configuring, initializing and starting simulation
- ▶ Model interface for OpenModelica Simulation Unit (OMSU) code generation is based on FMI 2.0 ME
- ▶ Additional Interface for
 - ★ Solver for algebraic loops
 - ★ Efficient access to model variables

Current Concept

OMSI: Simulation and Model Exchange Interface



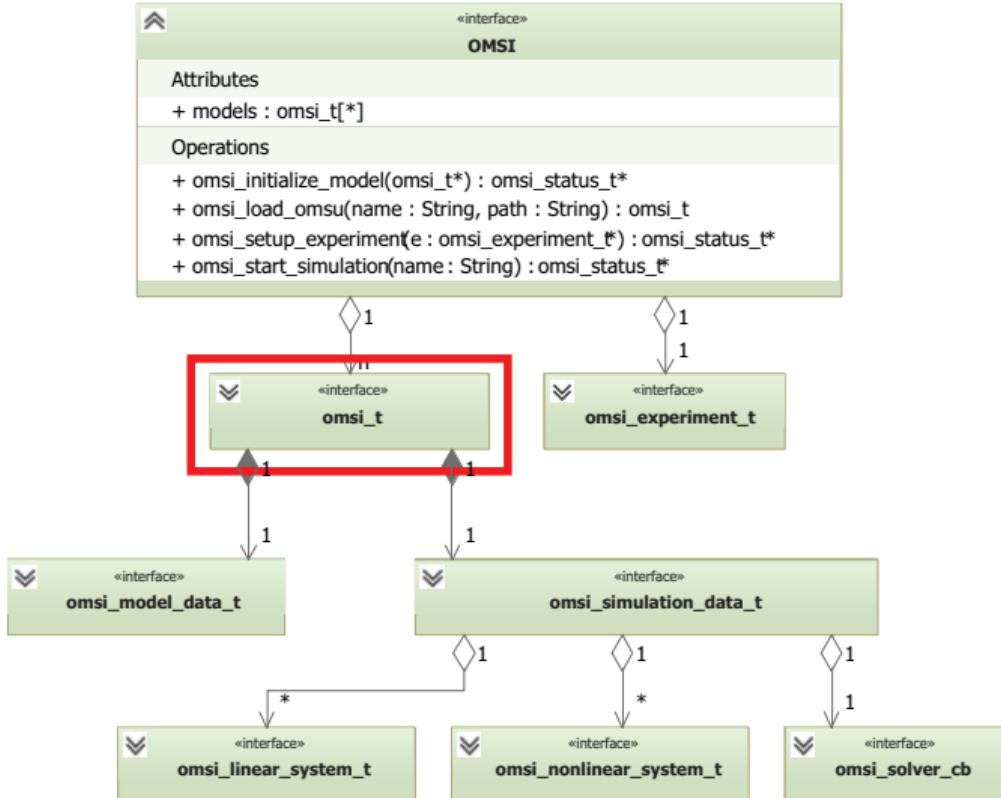
OMSI.h

- Functions for:

- ▶ Loading OMSU/FMU
- ▶ Configuring simulation
- ▶ Initializing simulation
- ▶ Starting simulation

Current Concept

OMSI: Simulation and Model Exchange Interface

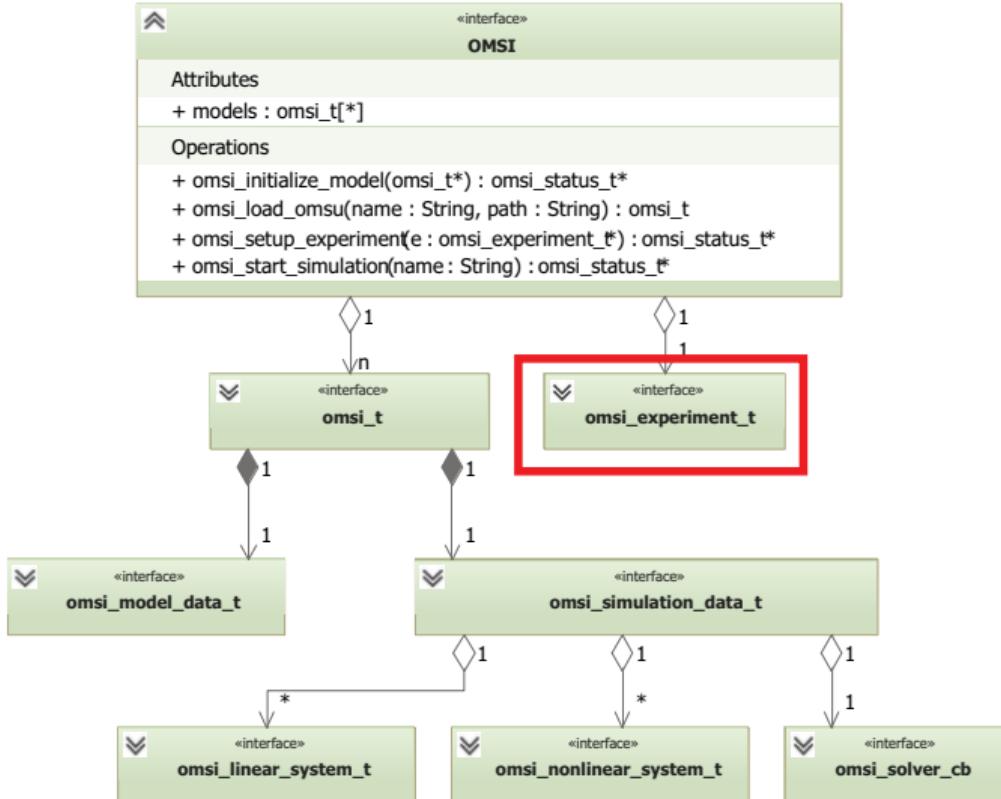


omsi_t

- Main interface of OMSI
- Contains data structures for model information and simulation data

Current Concept

OMSI: Simulation and Model Exchange Interface

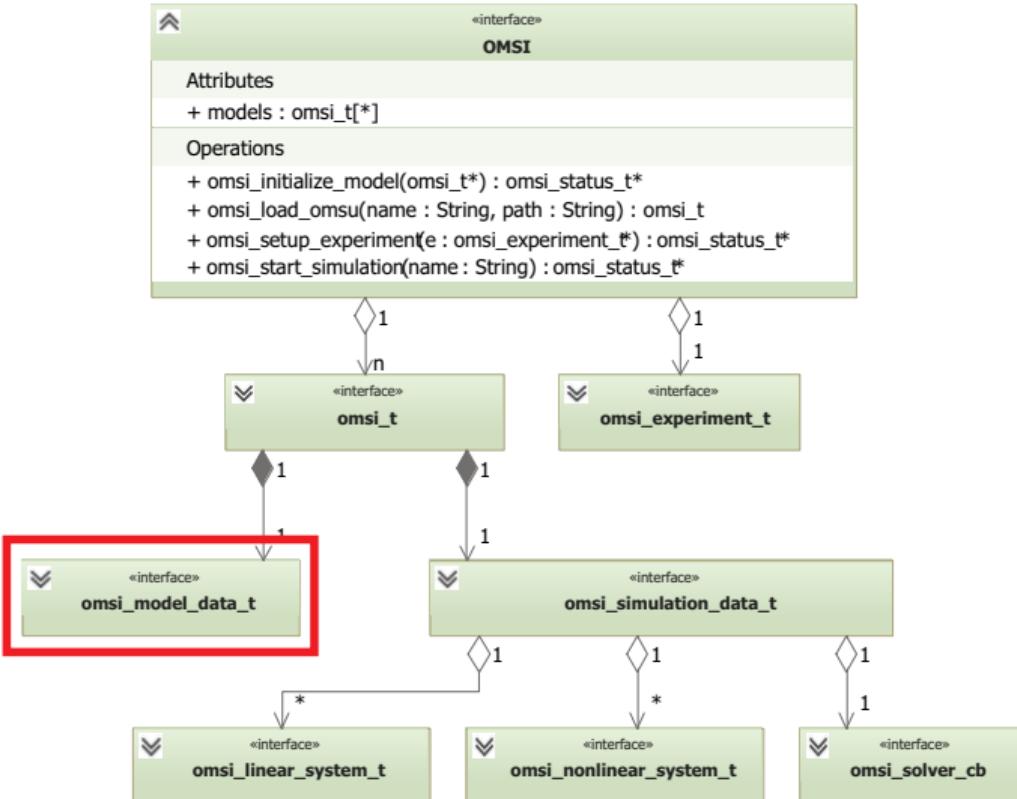


omsi_experiment_t

- Holds information for simulation settings like
 - ▶ Start time
 - ▶ Selected ODE solver
 - ▶ Selected solver for algebraic loops

Current Concept

OMSI: Simulation and Model Exchange Interface



omsi_model_data_t

- Holds model information like
 - ▶ Number of states
 - ▶ Number of real model variables
 - ▶ Number of zero crossings

Current Concept

OMSI: Simulation and Model Exchange Interface

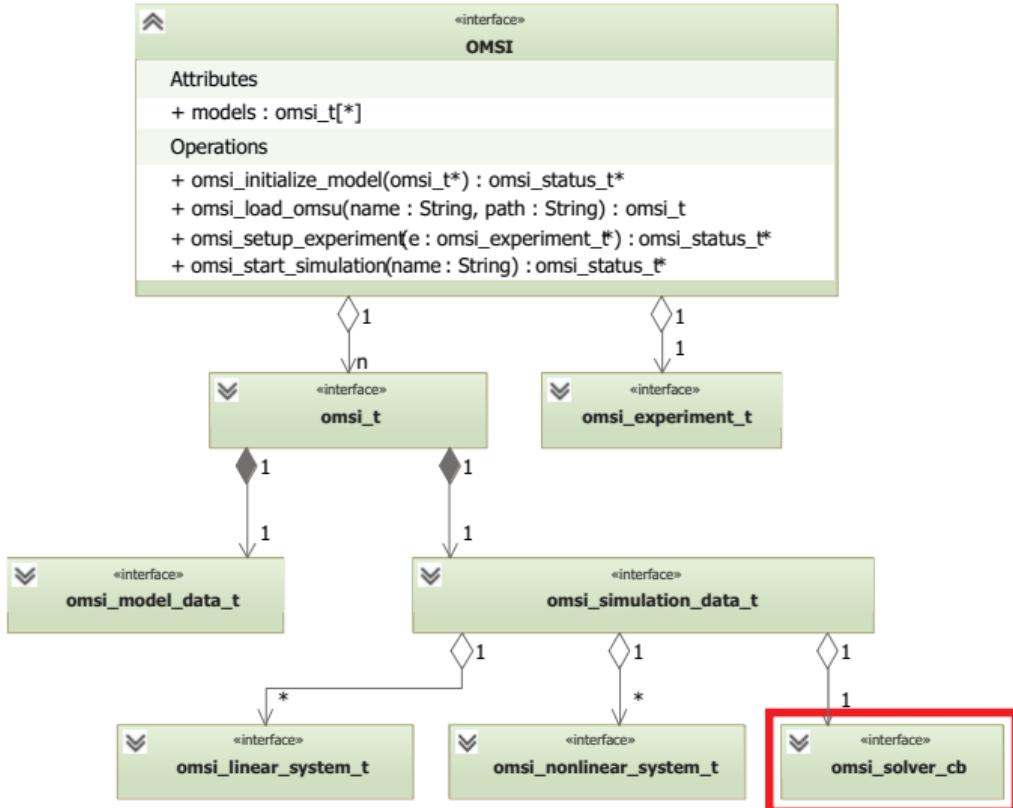


omsi_simulation_data_t

- Stores the simulation data like
 - ▶ Model variables
 - ▶ Model pre-variables
 - ▶ Zero crossing functions
 - ▶ Zero crossing conditions
- Holds systems for linear and non linear agloops

Current Concept

OMSI: Simulation and Model Exchange Interface

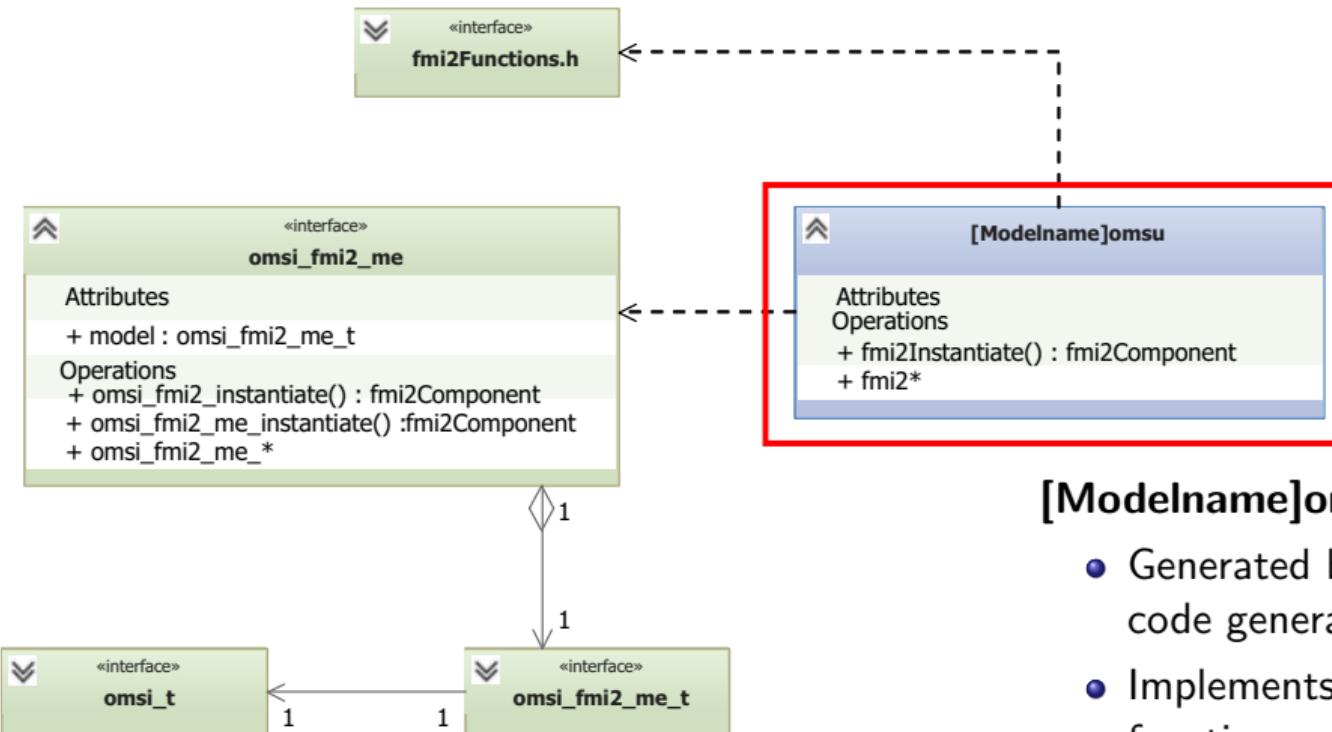


omsi_solver_cb

- Callback functions for solving algebraic loop systems

Current Concept

OMSI: Model Exchange based on FMI

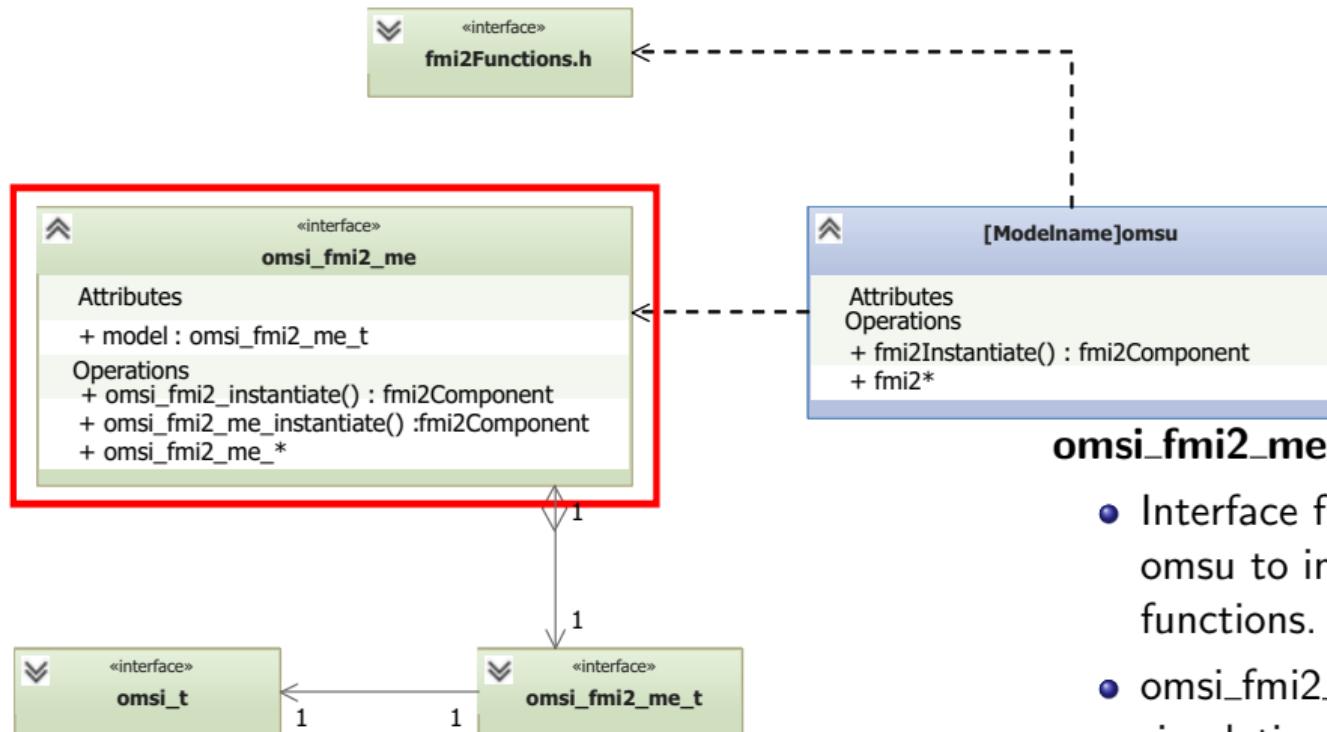


[Modelname]omsu

- Generated by the omc omsu code generation
- Implements FMI2.0 ME functions

Current Concept

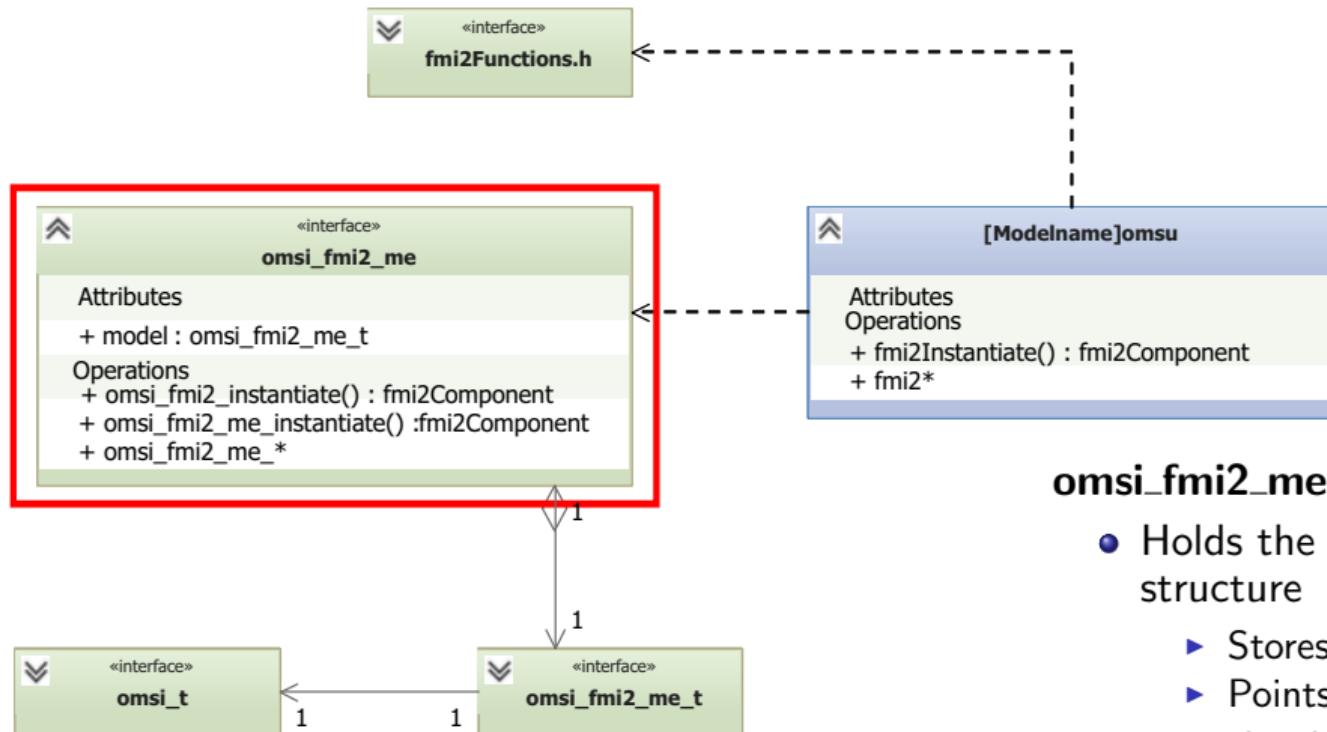
OMSI: Model Exchange based on FMI



- Interface functions called by omsu to implement FMI 2.0 functions.
- omsi_fmi2_me functions using simulation runtime functions

Current Concept

OMSI: Model Exchange based on FMI



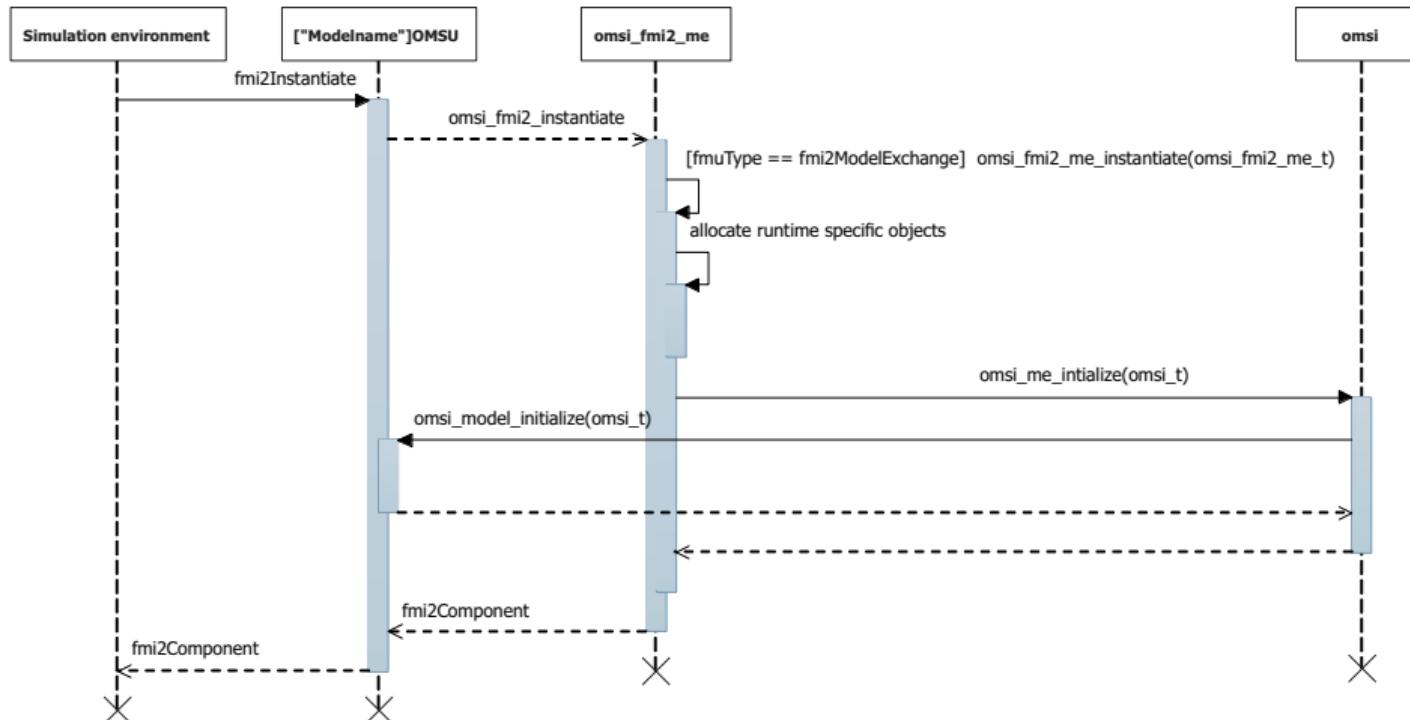
omsi_fmi2_me

- Holds the **omsi_fmi2_me_t** data structure
 - ▶ Stores specific FMI 2.0 data
 - ▶ Points to the **omsi_t** main data structure

Current Concept

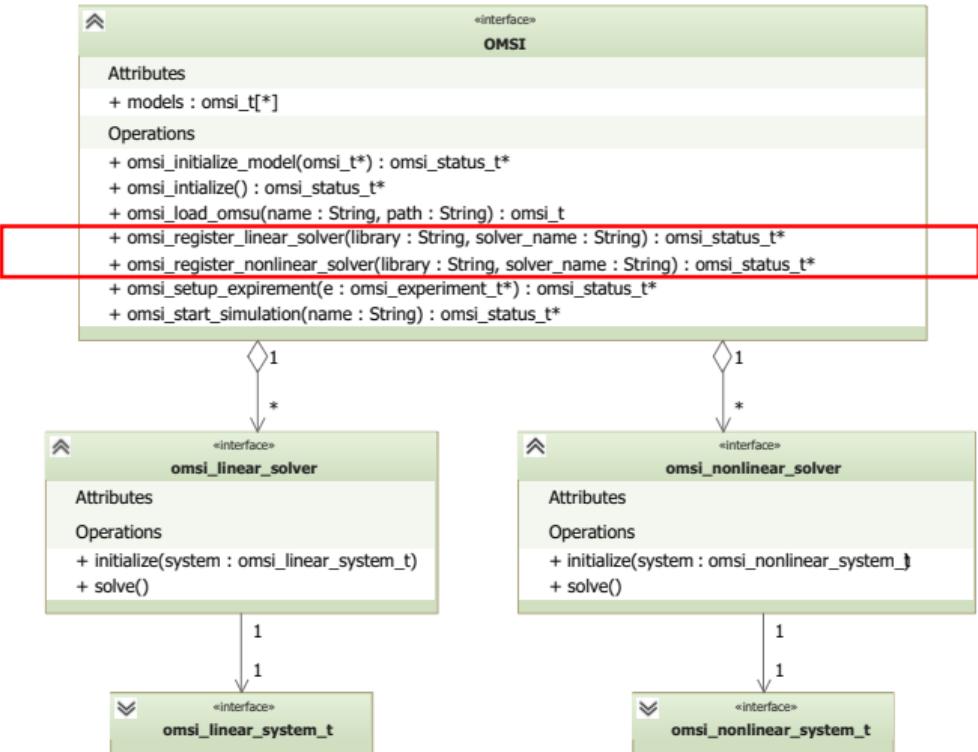
OMSI: Model Exchange based on FMI

Example for *fmi2Instantiate* call



Current Concept

OMSI: Solver Interface



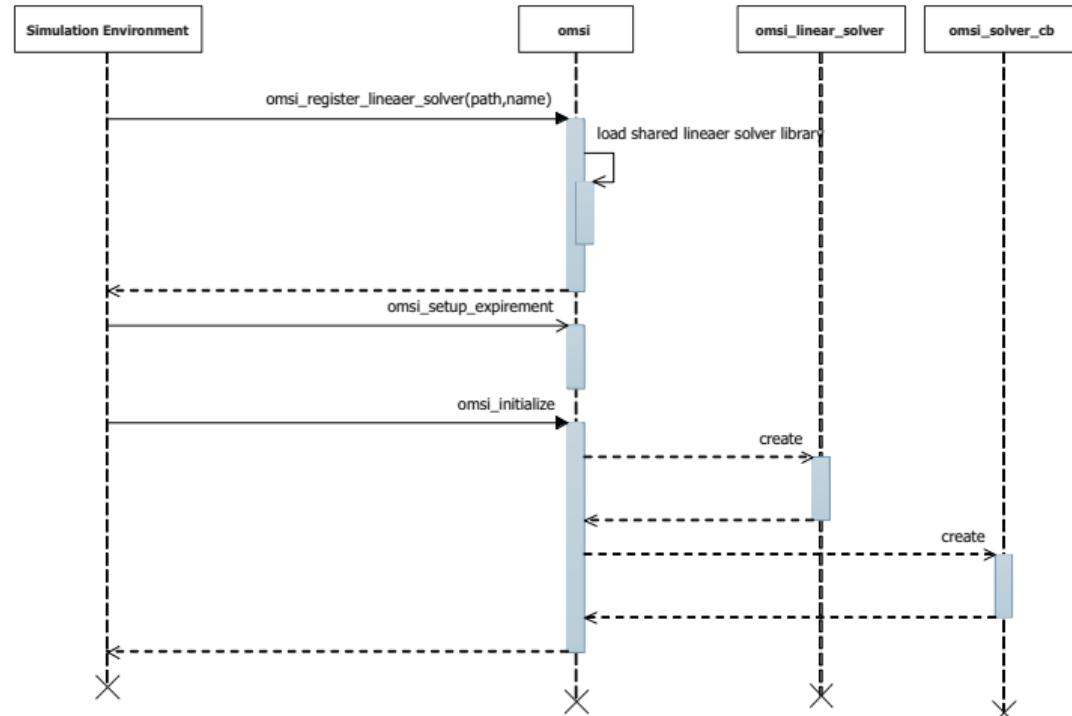
Linear and Nonlinear Solver

- Possible to register own linear or nonlinear solver for solving linear and nonlinear algebraic loops

Current Concept

OMSI:Solver Interface

Example for register linear solver



OMSI Model Interface:

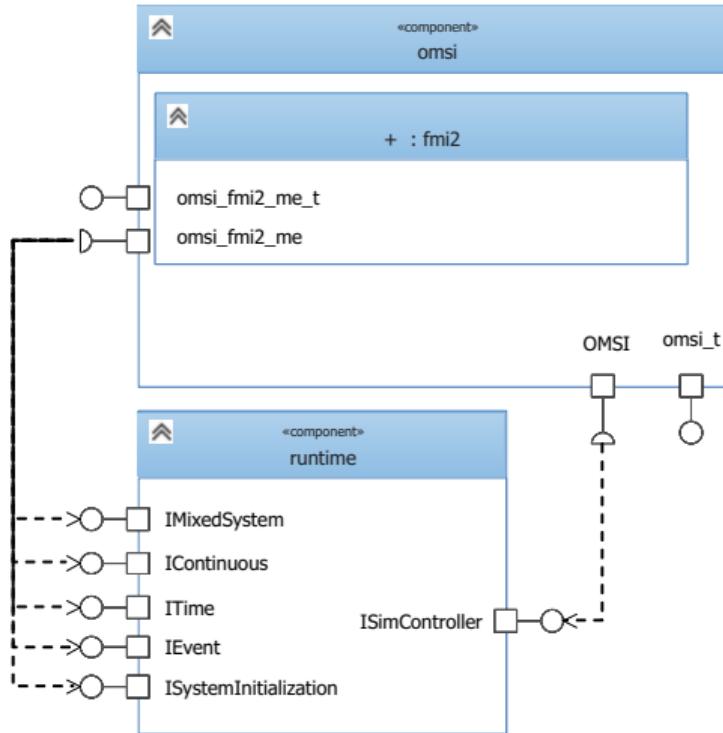
- New omsu code generation is based on the fmi cpp code generation
- omsu code generation implements the *omsi_fmi2_me* interface
- Open tasks:
 - ▶ Connection to omsi_t data interface
 - ▶ Support for linear and nonlinear solver interface

Current Statust

OMSI: Code Generation

OMSI Simulation Interface:

- Adapted cpp runtime for OMSI simulation interface
 - ▶ Separated in two components *omsi* and *runtime*
 - ▶ *omsi* using *runtime* to implement omsi functions



General open tasks:

- Adjust the SimCode and Template phase for the use of the OMSI data model
- Adjust the runtimes for OMSI simulation functions
- Build up general solver library

General open tasks:

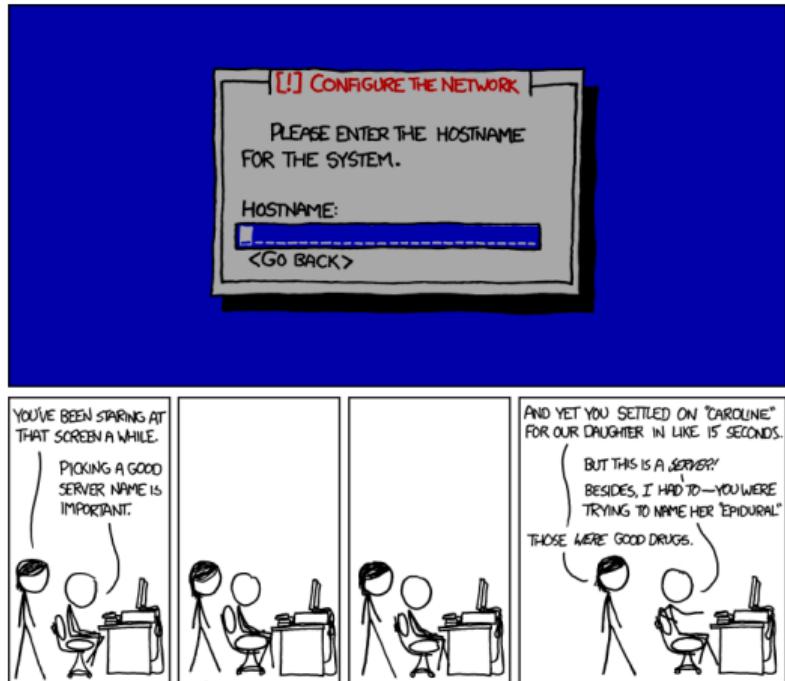
- Adjust the SimCode and Template phase for the use of the OMSI data model
- Adjust the runtimes for OMSI simulation functions
- Build up general solver library

Summary

- New Simulation and Model Interface based on FMI 2.0 ME
- Reduction of maintenance work, since fewer Code Generators are needed.
- New feature developments can be shared more easily.
- Since the interface is based on FMI, it is easily possible to integrate the generated omc model code in other projects

Summary

OMSI | OSMI | something else?



Credit: xkcd.com

General open tasks:

- Adjust the SimCode and Template phase for the use of the OMSI data model
- Adjust the runtimes for OMSI simulation functions
- Build up general solver library

Summary

- New Simulation and Model Interface based on FMI 2.0 ME
- Reduction of maintenance work, since fewer Code Generators are needed.
- New feature developments can be shared more easily.
- Since the interface is based on FMI, it is easily possible to integrate the generated omc model code in other projects

