Towards an unified OpenModelica Simulation Interface - OMSI

Current development status

Andreas Heuermann[1] Willi Braun[1] Niklas Worschech[2] Bernhard Bachmann[1]

FH Bielefeld University of Applied Sciences [1]
Bielefeld, Germany

Bosch Rexroth [2]
Lohr, Germany

February 2, 2019

SPONSORED BY THE

Federal Ministry
of Education
and Research

PARADOM

Outline

Motivation

Current issues

FH Bielefeld University of Applied Sciences

Rexroth Bosch Group

```
uniontype SimCode
  "Root data structure containing information required for
       templates to generate simulation code for a Modelica
       model."
  record SIMCODE
    ...
    list<DAE.Constraint> constraints;
    list<DAE.ClassAttributes> classAttributes;
    list<BackendDAE.ZeroCrossing> zeroCrossings;
    list<BackendDAE.ZeroCrossing> relations "only used by c
         runtime";
    list<BackendDAE.TimeEvent> timeEvents "only used by c runtime
          yet";
    list<DAE.ComponentRef> discreteModelVars;
    ExtObjInfo extObjInfo;
    SimCodeFunction.MakefileParams makefileParams;
    DelayedExpression delayedExps;
    list<JacobianMatrix> jacobianMatrixes;
    Option<SimulationSettings> simulationSettingsOpt;
    String fileNamePrefix, fullPathPrefix "Used in FMI where
         files are generated in a special directory";
    String fmuTargetName;
    HpcOmSimCode.HpcOmData hpcomData;
    ...
```

- SimCode data structure got bloated over time
- SimCode not independent of target language
- No clear separation between model and runtime data

FH Bielefeld University of Applied Sciences

Rexroth Bosch Group

C runtime

ModelicaTest_3.2.2 test using OpenModelica

Total	Frontend	Backend	SimCode	Templates	Compilation	Simulation	Verification
519	514	512	512	512	512	496	452

Total time taken: 1:32:57

System info: Intel(R) Core(TM) i7-6900K CPU @ 3.20GHz, 126 GB RAM, Ubuntu 18.04.1 LTS

OpenModelica Version: OMCompiler v1.14.0-dev.59+g683050ef1

Test started: 2019-01-19 22:19:07

Tested Library: 3.2.2

Cpp runtime

ModelicaTest_3.2.2_cpp test using OpenModelica

Total	Frontend	Backend	SimCode	Templates	Compilation	Simulation	Verification
519	514	512	512	512	508	475	369

Total time taken: 2:06:21

System info: Intel(R) Core(TM) i7-6900K CPU @ 3.20GHz, 126 GB RAM, Ubuntu 18.04.1 LTS

OpenModelica Version: OMCompiler v1.14.0-dev.59+g683050ef1

Test started: 2019-01-19 22:19:07

Tested Library: 3.2.2

- New features require development for each runtime

$\Rightarrow$ Different performance of runtimes

Goal: Make developer life more simple

- Unify data structures in back-end and simulation runtimes
- Share code generation for C and C++
- Clear interface and shared functionalities
  - Base: Use common base functionalities and solvers
  - C-Runtime: Simulation runtime in ANSI C for e.g. realtime applications
  - Cpp-Runtime: Simulation runtime for e.g desktop applications

Goal: Make developer life more simple

- Unify data structures in back-end and simulation runtimes
- Share code generation for C and C++
- Clear interface and shared functionalities
  - Base: Use common base functionalities and solvers
  - C-Runtime: Simulation runtime in ANSI C for e.g. realtime applications
  - Cpp-Runtime: Simulation runtime for e.g desktop applications

Goal: Make developer life more simple

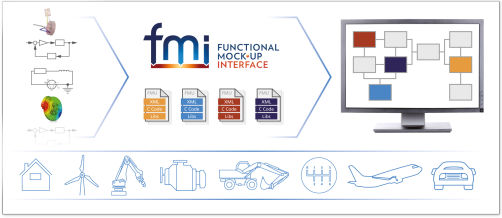
- Unify data structures in back-end and simulation runtimes
- Share code generation for C and C++
- Clear interface and shared functionalities
  - Base: Use common base functionalities and solvers
  - C-Runtime: Simulation runtime in ANSI C for e.g. realtime applications
  - Cpp-Runtime: Simulation runtime for e.g desktop applications

Goal: Make developer life more simple

- Unify data structures in back-end and simulation runtimes
- Share code generation for C and C++
- Clear interface and shared functionalities
  - Base: Use common base functionalities and solvers
  - C-Runtime: Simulation runtime in ANSI C for e.g. realtime applications
  - Cpp-Runtime: Simulation runtime for e.g desktop applications

Free extra

- Better FMI support



OpenModelica Simulation Interface (OMSI)

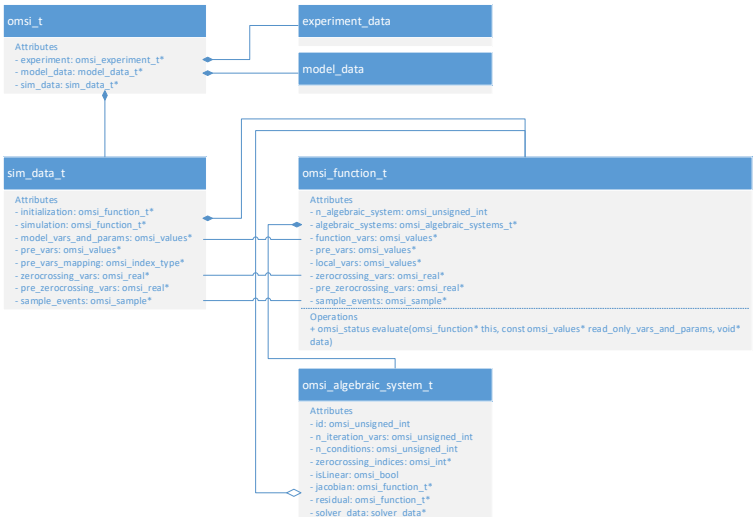
Outline

FH Bielefeld
University of
Applied Sciences

Rexroth
Bosch Group

Targets

- Separate data to smaller segments
- Store data thread-safe
- Forward only necessary informations
- Don't copy to much memory



- Include new OMSIData in SimCode

```
uniontype OMSIData
  "contains data for code generation for OMSI"
  record OMSI_DATA
    OMSIFunction initialization "contains equations and variables for initialization problem";
    OMSIFunction simulation "contains equations and variables for simulation problem";
  end OMSI_DATA;
end OMSIData;

uniontype OMSIFunction
  "contains equations and variables for initialization or simulation problem"
  record OMSI_FUNCTION
    list<SimEqSystem>        equations    "list of single equations and systems of equations";
    list<SimCodeVar.SimVar> inputVars    "list of simcode variables determining input variables for equation(s)";
    list<SimCodeVar.SimVar> outputVars   "list of simcode variables determining output variables for equation(s)";
    list<SimCodeVar.SimVar> innerVars    "list of simcode variables determining inner variables for equation(s), e.g $DER(x)";
    Integer nAllVars                     "number of input, inner and output vars";
    SimCodeFunction.Context context      "contains crefToSimVar hash table for lookup function in templates";
    Integer nAlgebraicSystems            "number of linear and non-linear algebraic systems in OMSI_FUNCTION.equations";
  end OMSI_FUNCTION;
end OMSIFunction;
```

- Shared functions for code generation

- Include new OMSIData in SimCode

```
uniontype OMSIData
  "contains data for code generation for OMSI"
  record OMSI_DATA
    OMSIFunction initialization "contains equations and variables for initialization problem";
    OMSIFunction simulation "contains equations and variables for simulation problem";
  end OMSI_DATA;
end OMSIData;

uniontype OMSIFunction
  "contains equations and variables for initialization or simulation problem"
  record OMSI_FUNCTION
    list<SimEqSystem>        equations    "list of single equations and systems of equations";
    list<SimCodeVar.SimVar> inputVars    "list of simcode variables determining input variables for equation(s)";
    list<SimCodeVar.SimVar> outputVars   "list of simcode variables determining output variables for equation(s)";
    list<SimCodeVar.SimVar> innerVars    "list of simcode variables determining inner variables for equation(s), e.g $DER(x)";
    Integer nAllVars                     "number of input, inner and output vars";
    SimCodeFunction.Context context      "contains crefToSimVar hash table for lookup function in templates";
    Integer nAlgebraicSystems            "number of linear and non-linear algebraic systems in OMSI_FUNCTION.equations";
  end OMSI_FUNCTION;
end OMSIFunction;
```

- Shared functions for code generation

- Include new OMSIData in SimCode

```
uniontype OMSIData
  "contains data for code generation for OMSI"
  record OMSI_DATA
    OMSIFunction initialization "contains equations and variables for initialization problem";
    OMSIFunction simulation "contains equations and variables for simulation problem";
  end OMSI_DATA;
end OMSIData;

uniontype OMSIFunction
  "contains equations and variables for initialization or simulation problem"
  record OMSI_FUNCTION
    list<SimEqSystem>        equations    "list of single equations and systems of equations";
    list<SimCodeVar.SimVar> inputVars    "list of simcode variables determining input variables for equation(s)";
    list<SimCodeVar.SimVar> outputVars   "list of simcode variables determining output variables for equation(s)";
    list<SimCodeVar.SimVar> innerVars    "list of simcode variables determining inner variables for equation(s), e.g $DER(x)";
    Integer nAllVars                     "number of input, inner and output vars";
    SimCodeFunction.Context context      "contains crefToSimVar hash table for lookup function in templates";
    Integer nAlgebraicSystems            "number of linear and non-linear algebraic systems in OMSI_FUNCTION.equations";
  end OMSI_FUNCTION;
end OMSIFunction;
```

- Shared functions for code generation

Generate some code

- Use common functions in templates
- Files for `omsi_function` and `omsi_alg_systems`
- Generate comparable code for equations

Generate some code

- Use common functions in templates
- Files for `omsi_function` and `omsi_alg_systems`
- Generate comparable code for equations

Generate some c

- Use common
- Files for omsi
- Generate con

helloWorldOMSI_omsic.c | helloWorldOMSI_sim_eqns.c | helloWorldOMSI_init_eqns.c

```
/* Evaluation functions for each equation */
/*
equation index: 3
type: SIMPLE_ASSIGN
der(x) = a * x
*/
void helloWorldOMSI_eqFunction_3(omsi_function_t* this_function, const omsi_values* model_vars_and_params){
  this_function->function_vars->reals[1] /* der(x) STATE_DER */
    = (model_vars_and_params->reals[2] /* a PARAM */) * (this_function->function_vars->reals[0] /* x STATE(1) */);
}


/* Equations evaluation */
omsi_status helloWorldOMSI_sim_eqns_allEqns(omsi_function_t* sim_eqns, omsi_values* model_vars_and_params, void* data){

  /* Variables */
  omsi_status status;

  status = omsi_ok;

  helloWorldOMSI_eqFunction_3(sim_eqns, model_vars_and_params);

  return status;
}
```

FH Bielefeld
University of
Applied Sciences

Rexroth
Bosch Group

Generate s

- Use co
- Files fo
- Genera

simpleNonLinLoop_0_init_eqns.c | simpleNonLinLoop_0_sim_eqns.c | simpleNonLinLoop_0_sim_eqns_algSyst_0. | simpleNonLinLoop_0_sim_eqns_derMat_0.c

```
}

/* Algebraic system evaluation */
/*
equation index: 10
type: ALGEBRAIC_SYSTEM
is linear: false
depending functions indices: 13, 12, 11
dimension: 3
iteration vars: b._SeedNLSJac3, a._SeedNLSJac3

c._$pDERNLSJac3._dummyVarNLSJac3 = (-a.SeedNLSJac3) - (-1.5) * b.SeedNLSJac3
$res1._$pDERNLSJac3._dummyVarNLSJac3 = 2.0 * (c * c.$pDERNLSJac3.dummyVarNLSJac3 + b * b.SeedNLSJac3 + a * a.SeedNLSJac3)
$res2._$pDERNLSJac3._dummyVarNLSJac3 = a.SeedNLSJac3 + b.SeedNLSJac3 + c.$pDERNLSJac3.dummyVarNLSJac3
*/
omsi_status simpleNonLinLoop_0_sim_eqns_algSystFunction_0(omsi_algebraic_system_t* this_alg_system,
        const omsi_values* model_vars_and_params, void* data){

  /* Variables */
  omsi_status status;

  /* Log function call */
  filtered_base_logger(global_logCategories, log_all, omsi_ok,
      "fmi2Evaluate: Solve algebraic system 0.");

  /* call API function something */
  status = omsi_solve_algebraic_system(this_alg_system, model_vars_and_params);

  return status;
}
```

Generate some code

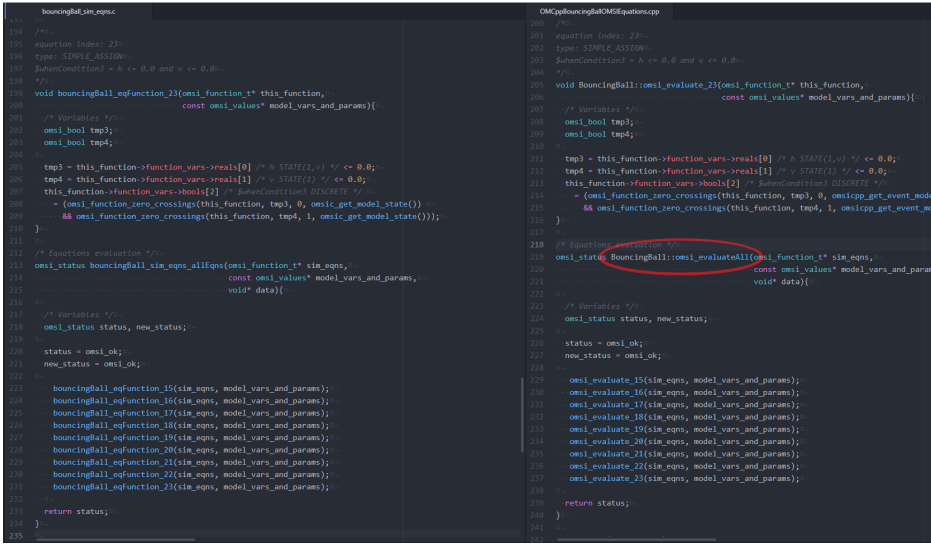
- Use common functions in templates
- Files for `omsi_function` and `omsi_alg_systems`
- Generate comparable code for equations

Generate some code

- Use common functions in templates
- Files for `omsi_function` and `omsi_alg_systems`
- Generate comparable code for equations

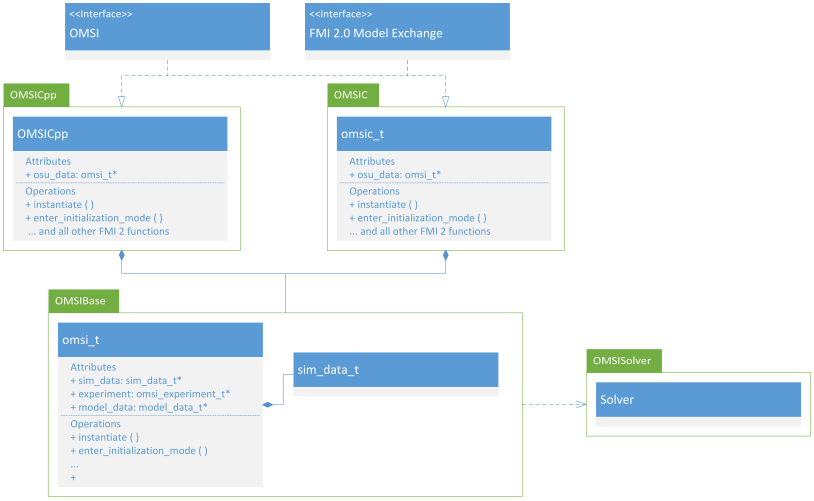
Overview

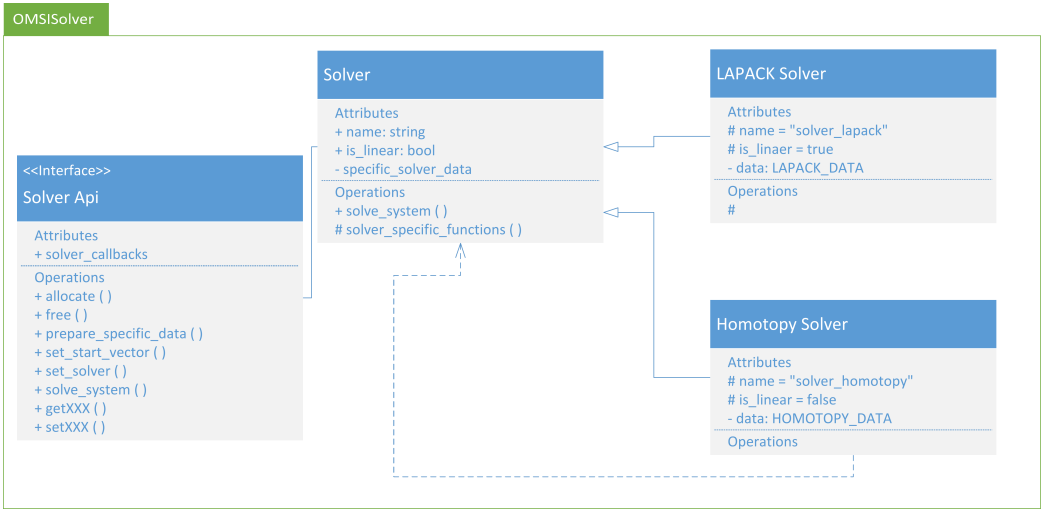
Templates



Overview

Call structure for OMSIC and OMSICpp



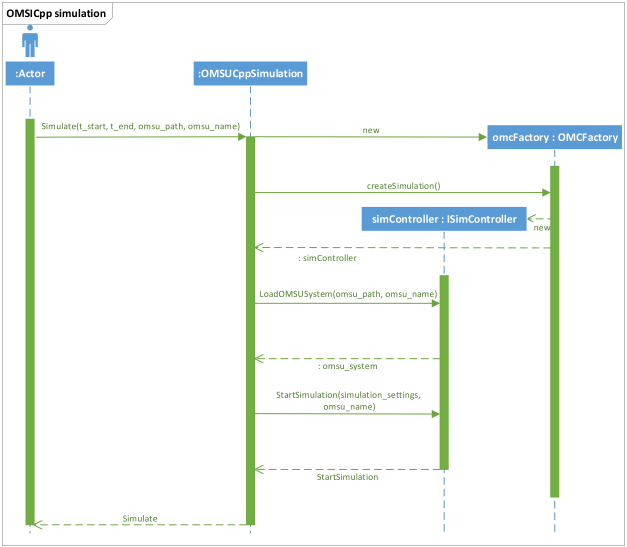

OMSISolver
Solver
Attributes
+ name: string
+ is_linear: bool
- specific_solver_data
Operations
+ solve_system ( )
# solver_specific_functions ( )
LAPACK Solver
Attributes
name = "solver_lapack"
is_linaer = true
- data: LAPACK_DATA
Operations
#
<<Interface>>
Solver Api
Attributes
+ solver_callbacks
Operations
+ allocate ( )
+ free ( )
+ prepare_specific_data ( )
+ set_start_vector ( )
+ set_solver ( )
+ solve_system ( )
+ getXXX ( )
+ setXXX ( )
Homotopy Solver
Attributes
name = "solver_homotopy"
is_linear = false
- data: HOMOTOPY_DATA
Operations

FH Bielefeld University of Applied Sciences

Rexroth Bosch Group

OMSU Cpp simulation runtime

- Simulate OMSU or FMI 2.0 Model Exchange FMU
- Optional arguments passed to simulation executable, e.g. experiment settings
- Aim to use OMSI ODE solver



Outline

What's working at the moment

- Code generation equations and equation systems
- Solving of linear and non-linear algebraic loops
- Event handling
- Exporting of Model Exchange FMU's

Modelica Standard Library - 3.2.2

- Build OMSIC FMU and import with OpenModelica to generate simulation executable
- Tested on Ubuntu Bionic (18.04)
- First models are building and simulating
- Just started, great improvements are to be expected

Next steps

- Complete code generation for all equations and equation systems
- Validate results automatically
- Test OMSICpp
- Merge first version into Modelica master
- Increase number of usable solvers on OMSISolver library

Outline

Summary

- Unified data structures in SimCode, Templates and runtimes
- Reduction of maintenance work
- Shared solver library
- Support of FMI 2.0 (ModelExchange)
- First working simulations

Open tasks

- Complete SimCode and Templates
- Reach high MSL coverage
- Swith from "old" C and Cpp runtime to OMSI runtime
- Exploit parallelism

Summary

- Unified data structures in SimCode, Templates and runtimes
- Reduction of maintenance work
- Shared solver library
- Support of FMI 2.0 (ModelExchange)
- First working simulations

Open tasks

- Complete SimCode and Templates
- Reach high MSL coverage
- Swith from "old" C and Cpp runtime to OMSI runtime
- Exploit parallelism

Thank you for your attention!

Summary

- Unified data structures in SimCode, Templates and runtimes
- Reduction of maintenance work
- Shared solver library
- Support of FMI 2.0 (ModelExchange)
- First working simulations

Open tasks

- Complete SimCode and Templates
- Reach high MSL coverage
- Swith from "old" C and Cpp runtime to OMSI runtime
- Exploit parallelism

Questions?