The OpenModelica Compiler BackEnd Milestones and Road Map

Jens Frenkel, TU Dresden
1. Introduction

2. Milestones

3. Road Map
1. Introduction

(Open)Modelica Compiler:

FrontEnd → BackEnd → Simulation Runtime

Modelica Model

Flat Modelica Model

Programm Code

Simulation

C/C++

model Pendulum
  parameter Real m = 0.5;
  parameter Real g = 9.82;
  parameter Real L = 1;
  Real x(start=L), y(start=0);
  Real xd,yd;
  Real Fo;

equation
  der(y) = yd;
  der(x) = xd;
  m*der(xd) = -(x*Fo)/L;
  m*der(yd) = -m*(g-(Fo*y))/L;
  x*x + y*y = L^2;
end Pendulum;
1. Introduction

(Open)Modelica Compiler:

- Parsing
- Replacements
- Modifications
- Instantiation
- Lookup

FrontEnd

BackEnd

- Causalization
- Indexreduction
- Simplification
- Solvability

Simulation Runtime

- Numerical Computation

Modelica Model

Flat Modelica Model

Programm Code

Simulation
1. Introduction

Quality:
- Supported Language Features
- Robustness
- Reliability
- Efficiency
- Helpfulness of Error Messages
1. Introduction

Quality

Usable

Unusable

FrontEnd

BackEnd

Simulation Runtime

Compiler Phases

Quality dimensions for the OpenModelica Compiler BackEnd Milesstones and Road Map.
1. Introduction

Target Groups:
- Modelica Developers
- Compiler
- Library
- Modelica Users
- Commercial
- Scientific
- Modelica Starters

OpenModelica have to meet increasing requirements

FrontEnd  BackEnd  Simulation Runtime  Compiler Phases

Quality
1. Introduction

Quality

FrontEnd | BackEnd | Simulation Runtime

End of 2010

Future OMC

OMC Trunc

Dymola

Compiler Phases

Linköping. 04/02/2013
2. Milestones

Problems - BackEnd - End of 2010

- Cluttered Program code
- Heuristic State Selection
- Basic support for Array Equations
- No Support for Complex, IF Equations
- Basic Simplification/Optimization
- Bad Scaling $O(N^3)$ -> Why?
2. Milestones

**Pendulum - translation**

- **Time [s]**
- **Number of Bodies**

- Causalization
- timetotal
2. Milestones

Backend Pipeline

Modelica Model

Pre Phase
Evaluate Parameter
Expand der Operator
Remove Simple Eqs.

Causalization Phase

Post Phase
semiLinear
Remove All Simple Eqs
Tearing

Output Phase
2. Milestones

Backend Pipeline

• High Flexibility
• Configurable via Flags and Script
• Accelerate Compiler Development
• Simplifies Implementation, Test and Comparison of Modules
• Now 16 PrePhase and 25 PostPhase Modules
2. Milestones

Restructuring of Program Code

• More fine grained package structure
  • More than 30 packages now
• Improve Maintainability
• Enhance Developing Process
2. Milestones

Support more Language Features in Data Structure

- Complex Equations
- If-Equations

Reimplement Support of Language Features

- Array Equations
- Event Handling
2. Milestones

Tearing

Incidence Matrix with Solvability

Unsolvable Vars -> Tearing Vars

Match Equations with one Var.

Select Tearing Variable

Torn System

Heuristic Based Selection:

• Unsolvable
• Wights on: Solvability, unassigned Edges, Type
2. Milestones

Tearing

• Implement robust and efficient Tearing Algorithm
• Take care on Solvability (no Division by Zero)
• Used by default
  • nonlinear System
• Usable by Flag
  • Linear Systems
  • Mixed Systems

-> Volker Waurich (TU Dresden):
  Comparison of Tearing Algorithms
2. Milestones

Remove Simple Equations

• \( a = b \)
• \( c = \text{constant} \)

• Almost linear scaling algorithm implemented
• Detect Singular Subsystems
2. Milestones

Matching

- **Survey of Matching Algorithms**
- **9 different Matching Algorithms usable**
  - Implemented in MetaModelica and C
    ```
    setMatchingAlgorithm("PFPlusExt");
    {BFSB,DFSB,MC21A,PF,PFPLUS,HK,HKDW,ABMP,PR,
    BFSBExt,DFSBExt,MC21AExt,PFExt,PFPLUSExt,HKExt,HKDWExt,ABMPExt,PRExt}
    ```
  - different heuristic based Matching Algorithms
    - Implemented in MetaModelica (2) and C (3)
    - speed up
      ```
      setCheapMatchingAlgorithm(3); {0,1,2,3}
      ```
- **Decrease Compilation Time**
2. Milestones

Causalization

- **PrePhase**
  - Singular System?
    - Index Reduction
      - Singular Subsystems?
        - State Selection
          - BLT Form
            - **PostPhase**

- **Match the system, distinguish not between state and state derivative**
- **Continue to use matching by remove assignments from States**
2. Milestones

Causalization

PrePhase

Singular System?

Index Reduction

Singular Subsystems?

State Selection

BLT Form

PostPhase

- Usable with all Matching Algorithms
- allow higher derivatives, der(x,3)
  - no additional memory
- Consider State Order, w=der(\(\phi\))
  - less Dummy Der. Variables
- Handle undifferentiable Subsets from StateSelect.prefer selection
2. Milestones

Causalization

PrePhase

Singular System?

Index Reduction

Singular Subsystems?

State Selection

BLT Form

PostPhase

\[
R1.i + R2.i - L1.i = 0.0 \\
L2.i + (-R2.i) - R1.i = 0.0
\]

\[
\begin{bmatrix}
1 & 1 & * \\
-1 & -1 & 1
\end{bmatrix}
\begin{bmatrix}
R1.i \\
R2.i
\end{bmatrix} =
\begin{bmatrix}
-L1.i \\
L2.i
\end{bmatrix}
\]

\[
\det((1*-1)-(-1*1)) = 0 \\
L2.i - L1.i = 0.0
\]
2. Milestones

Causalization

PrePhase

Singular System?

Index Reduction

Singular Subsystems?

State Selection

BLT Form

PostPhase

- Based on Selection Algorithm by Mattsson and Söderlind
- Improved to consider StateSelection
- Avoid Algebraic Loops by Selection of States (heuristic)
Symbolic Initialization

Lennart Ochel, Bernhard Bachmann, Willi Braun (FH Bielefeld):

Initialization within OpenModelica
2. Milestones

Pendulum - translation

Time [s] vs Number of Bodies

- Causalization
- timetotal
- Causalization New
- timetotal New
3. Road Map

Quality

OMC – Feature

OMC – Trunc

Compiler Phases

Quality

FrontEnd
BackEnd
Simulation Runtime

OMC – Feature

OMC – Trunc

Compiler Phases
3. Road Map

Tearing
• finalize efficient support for linear systems

Casualization
• improve detection of singular subsystems
• more efficient pivoting for dynamic state selection
• algorithm based selection of states to avoid algebraic loops
• support solvable singular subsystems
Symbolic Initialization

• improve support for unbalanced systems

Simulation Runtime

• Robustness
• Efficiency
• Decrease C/C++ Compilation Time
»Wissen schafft Brücken.«

Jens Frenkel
Dresden University of Technology
jens.frenkel@tu-dresden.de
http://tu-dresden.de/bft