A Modular OpenModelica Compiler Backend

J. Frenkel • W. Braun
A. Pop • M. Sjölund
Outline

1. Introduction

2. Concept of Modular Compiler Backend

3. Roadmap and Implementation Status

4. Conclusion
OpenModelica:
- open-source Modelica-based modelling and simulation environment for industrial and academic usage

- academic → research and teaching
Started in 2008 with OpenModelica as a user
Since 2009 member of the development team

- Bugfixes
  - Multibody
- Implemented new features
  - Tearing
  - Relaxation
  - Multibody
1. Introduction

What is planned for the backend?
- Extend existing algorithms
- Implemented new algorithms

What is wrong with the backend?
- Functions for symbolic manipulation are very low level
  - Direct access to basic structures
  - Consistency of equation system cannot be guaranteed
  - Changes lead to unexpected side effects

=> High level symbolic manipulation is needed
1. Introduction

Optimised form of the system of equations

- analyse
- evaluate

- implement
- test

→ ambitious task
→ needs support from development environment
→ clear, easy to understand and task related framework
1. Introduction

- Expectations
- Current Situation

Redesign of the compiler Backend
2. Concept

Packet Concept for Each Level

Symbolic Math

Equation System

Expression Solve

Symbol Simplify

Equation Algorithm

Variable

DAE-Handler for Index Reduction

Matching

Code Writer

Modules for Optimisation

Sorting...

Backend Implementation

Linköping, 07/02/2011 A Modular OpenModelica Compiler Backend Slide 8
2. Concept

Flattening Model

Input Phase

Pre-Optimisation

Transformation Phase

Past-Optimisation

Output Phase
2. Concept

Flat Model

Input Phase

Pre-Optimisation

Transformation Phase

Past-Optimisation

Output Phase

Evaluate Parameter

Remove Alias Equations

Function Inlining

...
2. Concept

- Flat Model
  - Input Phase
    - Pre-Optimisation
      - Transformation Phase
        - Past-Optimisation
          - Output Phase

- Matching
  - DAE-Handler for Index Reduction:
    • Dummy Derivative with Dynamic State Selection
    • Index Reduction and Projection/Stabilization

- Sorting
2. Concept

- Remove Alias Equations
- Inline Integration
- Function Inlining

Flat Model

Input Phase

Pre-Optimisation

Transformation Phase

Past-Optimisation

Output Phase
2. Concept

- Flat Model
  - Input Phase
    - Pre-Optimisation
      - Transformation Phase
        - Past-Optimisation
          - Output Phase
2. Concept

Assemble your own Compiler!

- evaluate parameter
- Remove Alias Equations
- Inline Integration
- Tearing Relaxation
- Function inlining
- ...
2. Concept

Interfaces

User Module

Flat Model

Input Phase

Pre-Optimisation

Transformation Phase

Past-Optimisation

Output Phase

External symbolic Equation system Optimisation Tool

... Output

... Output

XML File

Linköping, 07/02/2011
3. RoadMap/Status

1. Resort functions

2. Combine functions with the same purpose

3. Implement internal Interfaces for Equation System Pipeline

4. Improve performance of Equation System Pipeline

5. Improve existing Optimisation modules
3. RoadMap/Status

1. Resort functions ← DONE
2. Combine functions with the same purpose ← DONE
3. Implement internal Interfaces for Equation System Pipeline
4. Improve performance of Equation System Pipeline
5. Improve existing Optimisation modules

  Backend Reorganisation Week
new Backend Implementation:

- increase the clarity of the code
- decrease the error-proneness
- improve the extensibility
- improve the maintainability
- speed development process up

• improve compiler performance and dependability
• simplifies implementation of new features
• decrease barrier of becoming a compiler developer
• increases usability of the compiler
The Motion Platform at Dresden University can be visited before/during/after the Modelica Conference 2011 in Dresden.
»Wissen schafft Brücken.«

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