TECHNISCHE UNIVERSITÄT DRESDEN
Chair of Construction Machines and Conveying Technology

OpenModelica Workshop 2011

A Modular OpenModelica Compiler Backend

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

Outline

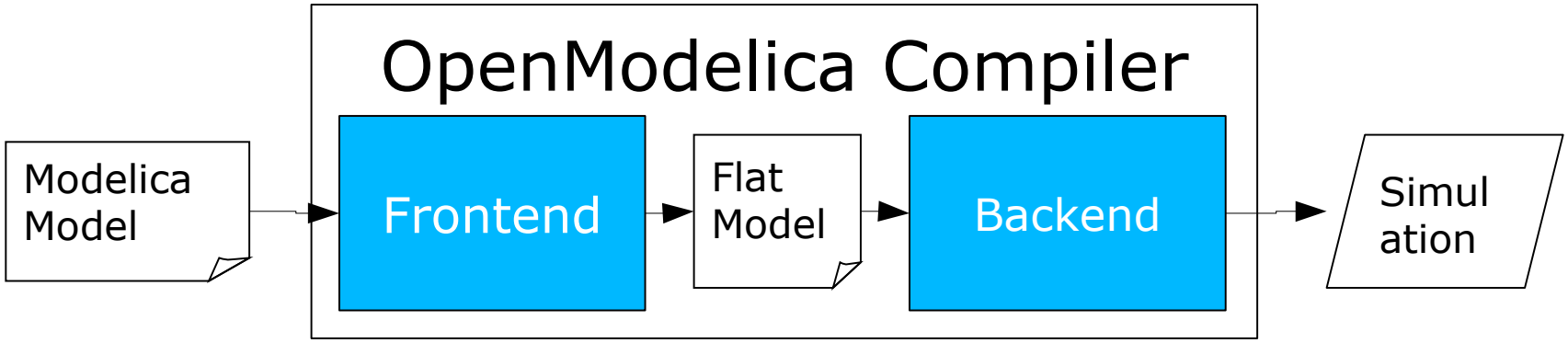
1. Introduction
2. Concept of Modular Compiler Backend
3. Roadmap and Implementation Status
4. Conclusion

1. Introduction

OpenModelica:

- open-source Modelica-based modelling and simulation environment for industrial and academic usage

- academic → research and teaching



1. Introduction

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

} system of equations

- implement
- test

} symbolic manipulation/optimisation algorithm

→ ambitious task

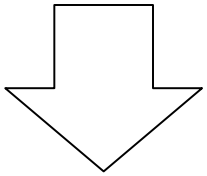
→ needs support from development environment

→ clear, easy to understand and task related framework

1. Introduction

Expectations

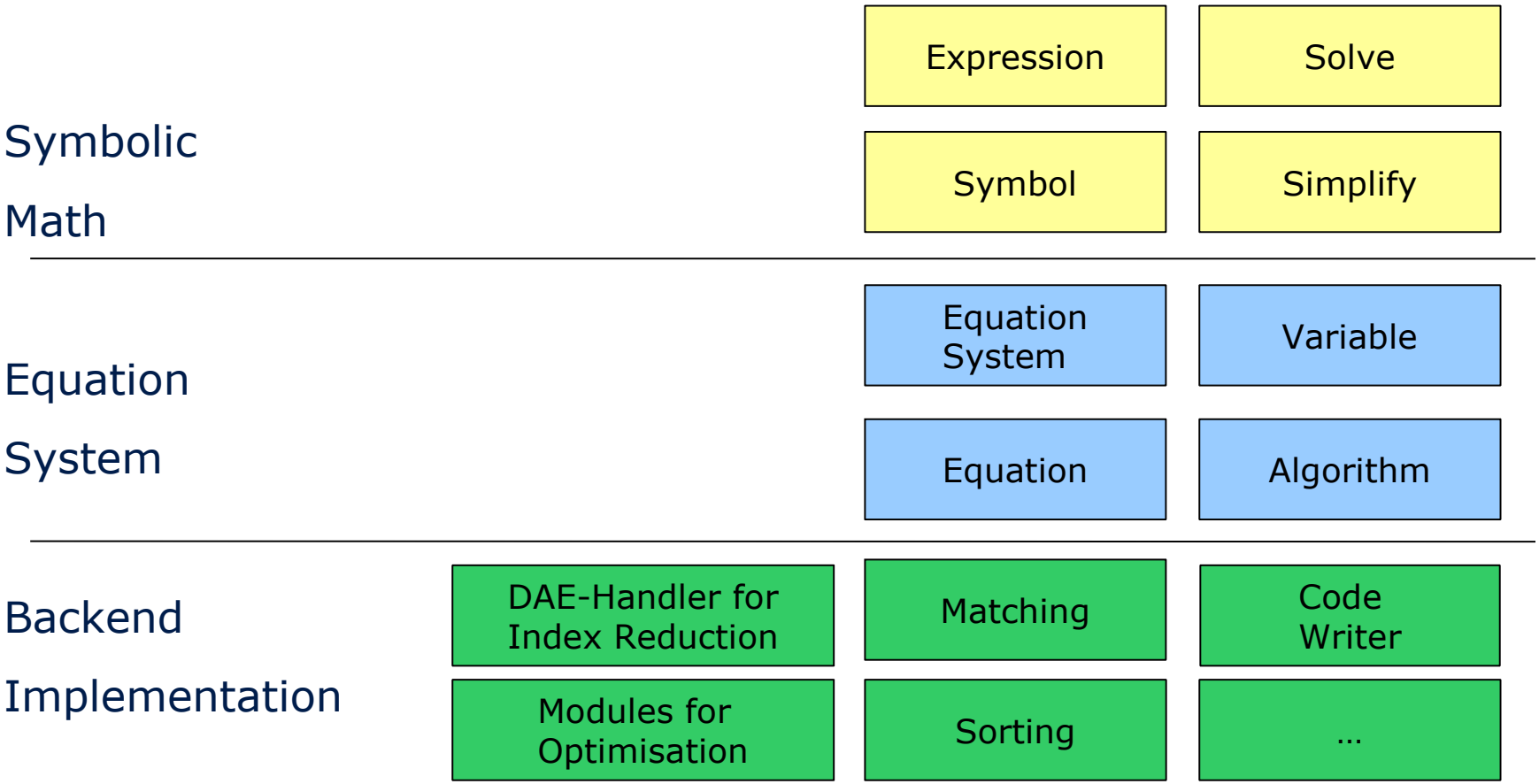
Current Situation



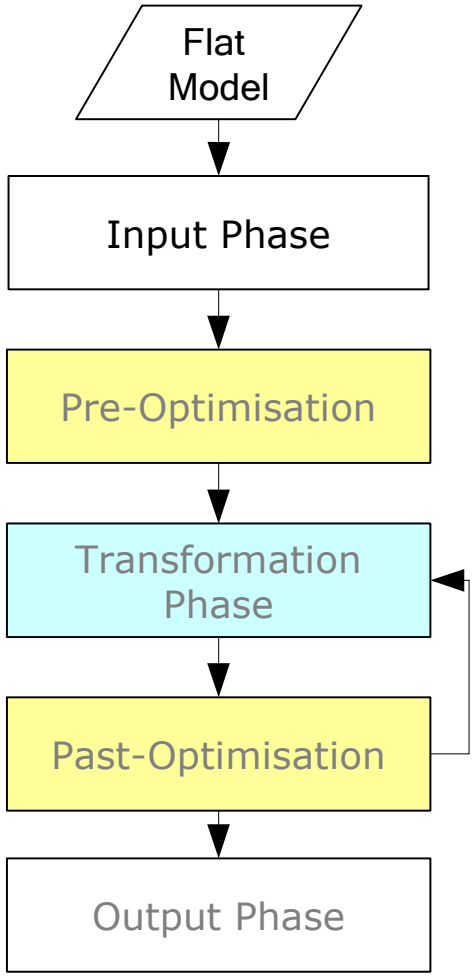
Redesign of the compiler Backend

2. Concept

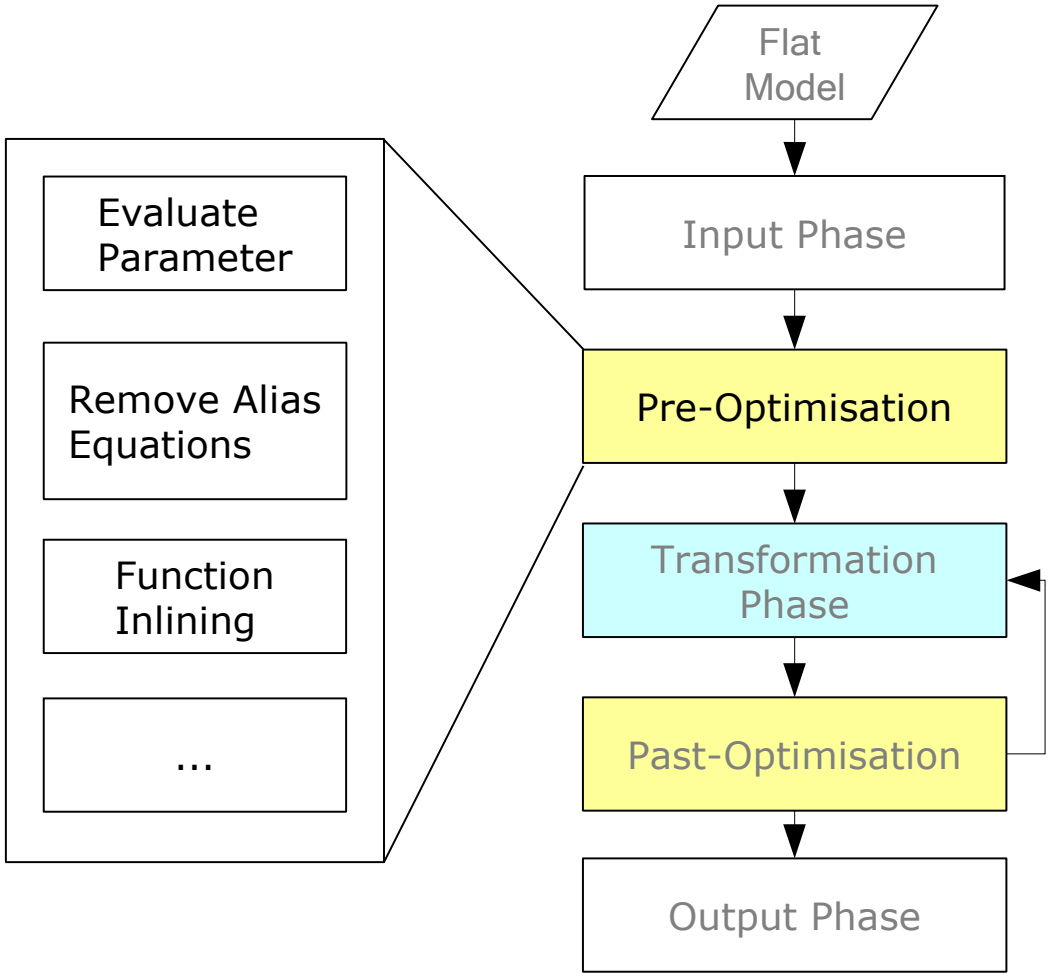
Packet Concept for Each Level



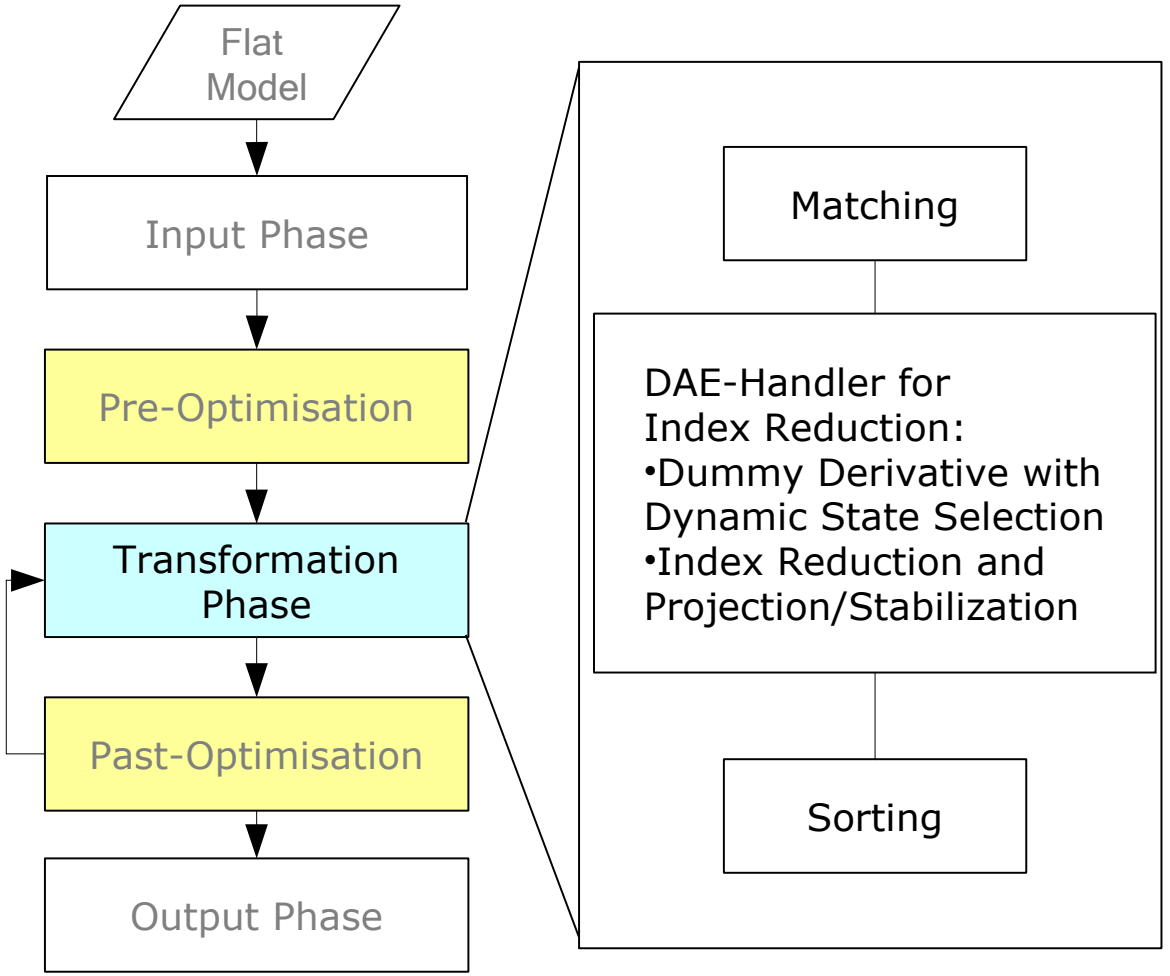
2. Concept


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

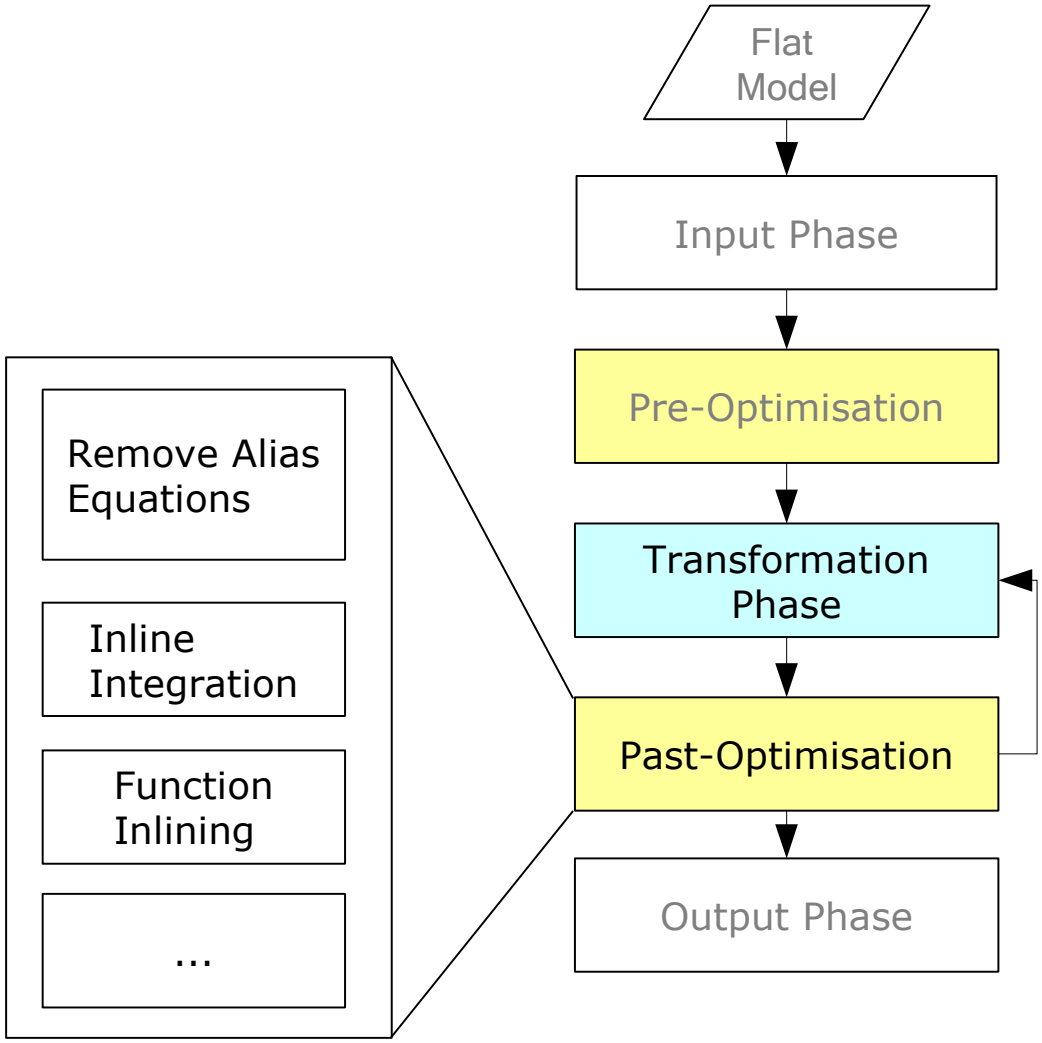
2. Concept


Flat Model
Input Phase
Pre-Optimisation
Transformation Phase
Past-Optimisation
Output Phase
Evaluate Parameter
Remove Alias Equations
Function Inlining
...

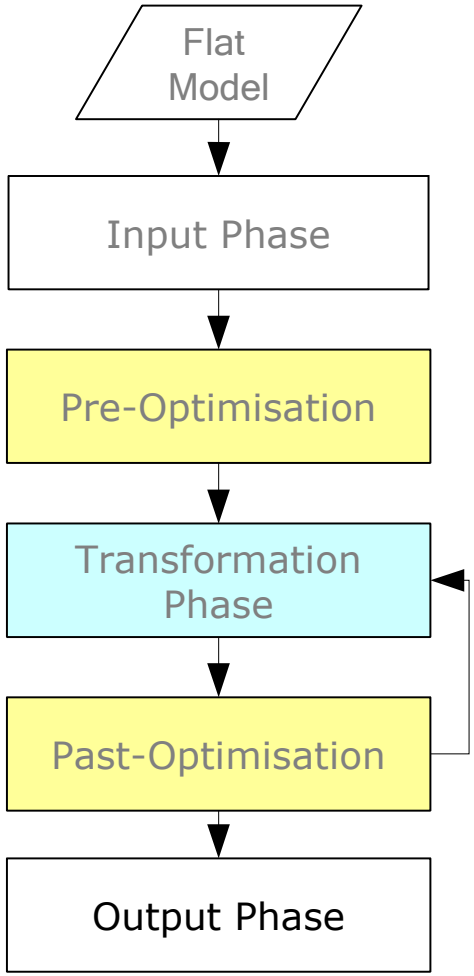
2. Concept


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

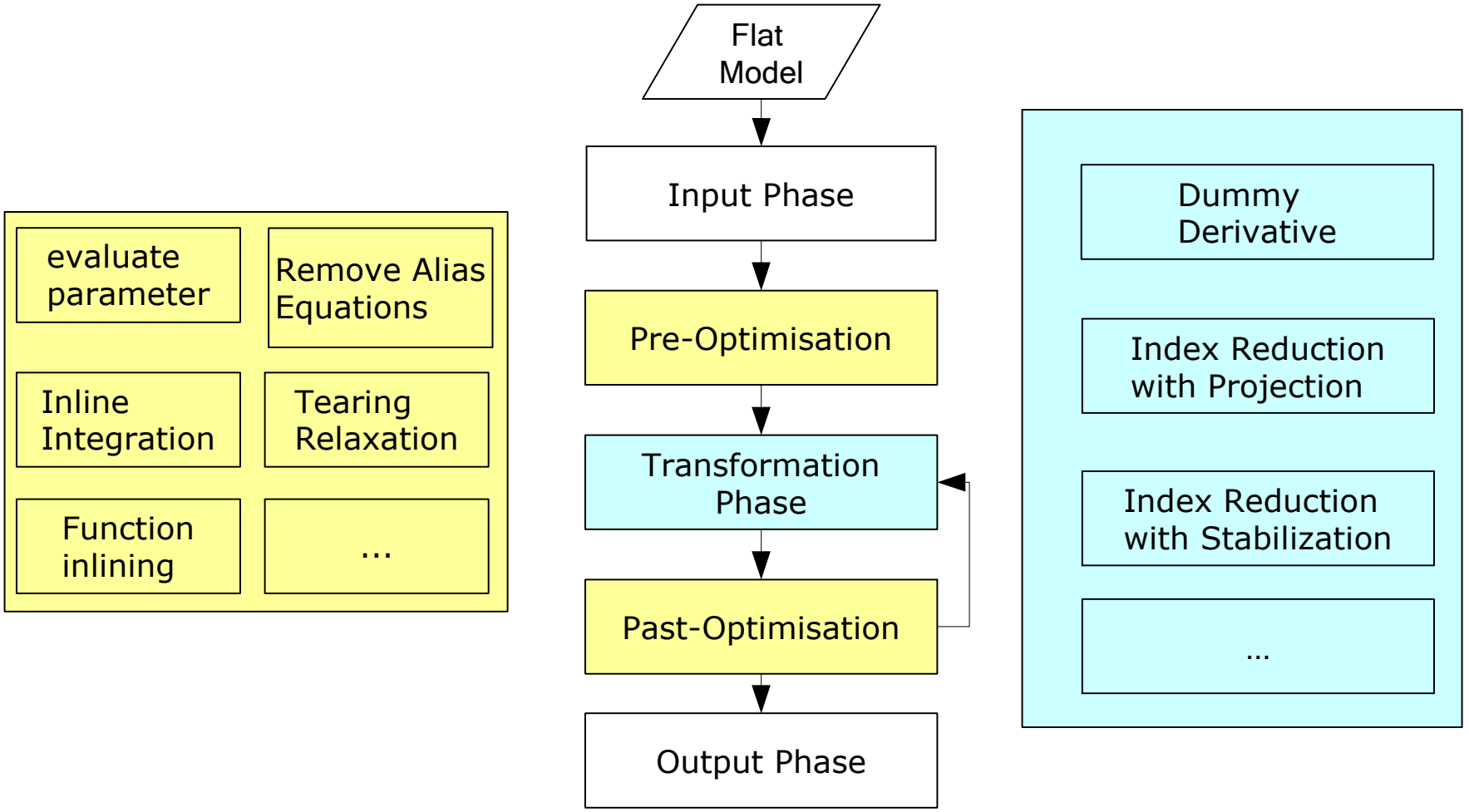

Flat Model
Input Phase
Pre-Optimisation
Transformation Phase
Past-Optimisation
Output Phase
Remove Alias Equations
Inline Integration
Function Inlining
...

2. Concept


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

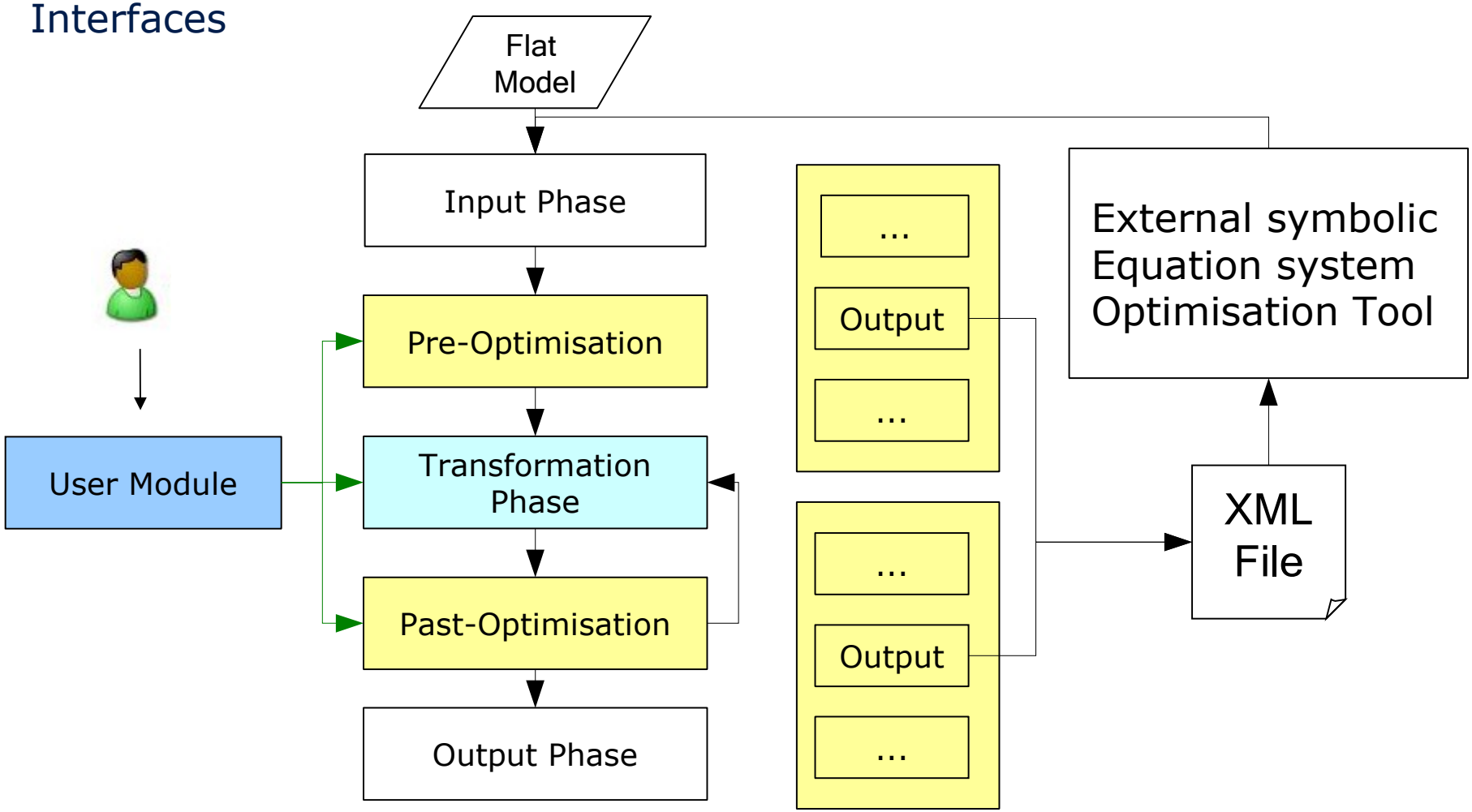
2. Concept

Assemble your own Compiler!



2. Concept

Interfaces



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

- From 24. – 30. Oct. 2010 at Linköping University Backend Reorganisation Week

4. Conclusion

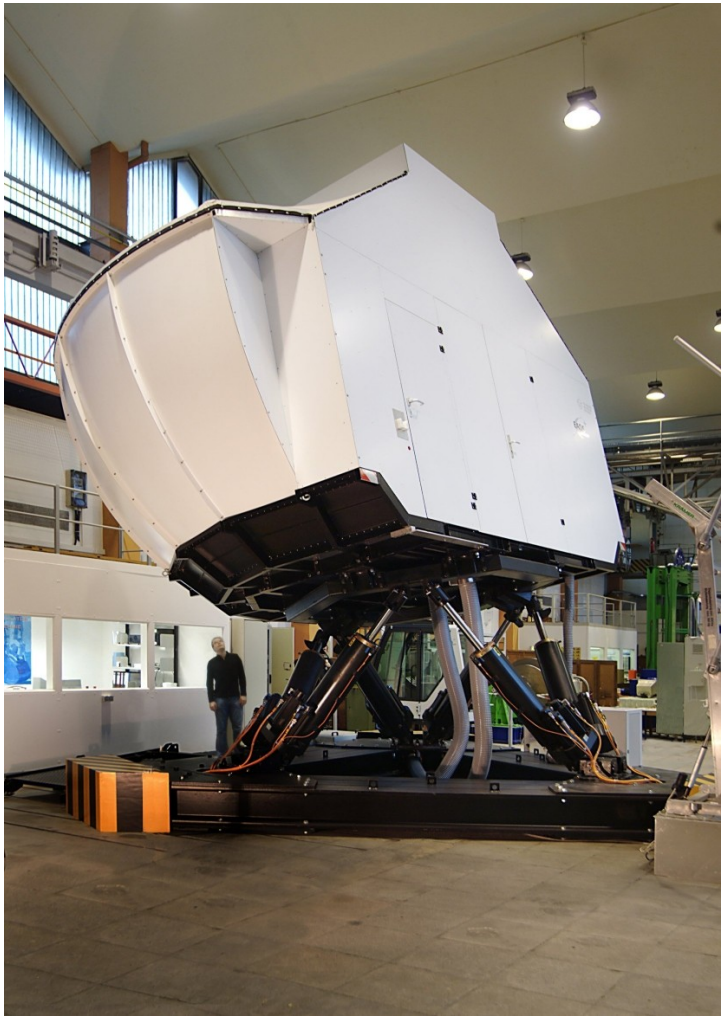
new Backend Implementation:

- 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 End

The Motion Platform at Dresden University can be visited before/during/after the Modelica Conference 2011 in Dresden.



The End

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