



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

OpenModelica Workshop 2013

Chair of Construction Machines and Conveying Technology

The OpenModelica Compiler BackEnd Milestones and Road Map

Jens Frenkel, TU Dresden

1. Introduction

2. Milestones

3. Road Map

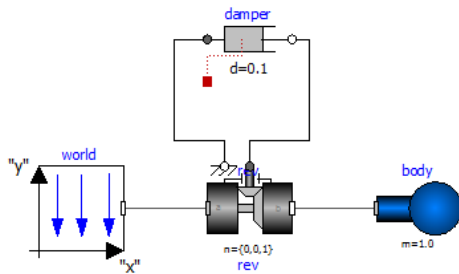
1. Introduction

(Open)Modelica Compiler:

FrontEnd

BackEnd

Simulation
Runtime

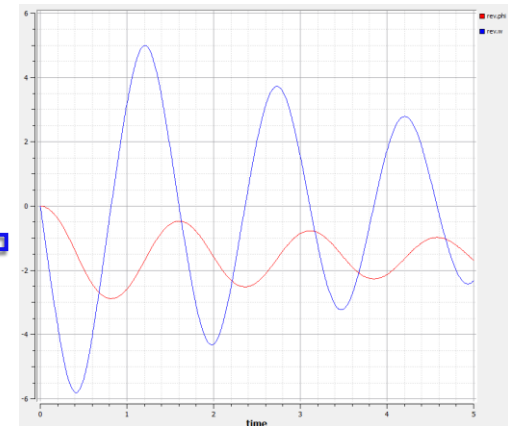


```

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;

```

C/C++



Modelica
Model

Flat
Modelica
Model

Programm
Code

Simulation

(Open)Modelica Compiler:



- Parsing
- Replacements
- Modifications
- Instantiation
- Lookup

Modelica
Model



- Causalization
- Indexreduction
- Simplification
- Solvability

Flat
Modelica
Model



- Numerical
Computation

Programm
Code

Simulation

1. Introduction

Quality

Quality:

- Supported Language Features
- Robustness
- Reliability
- Efficiency
- Helpfulness of Error Messages

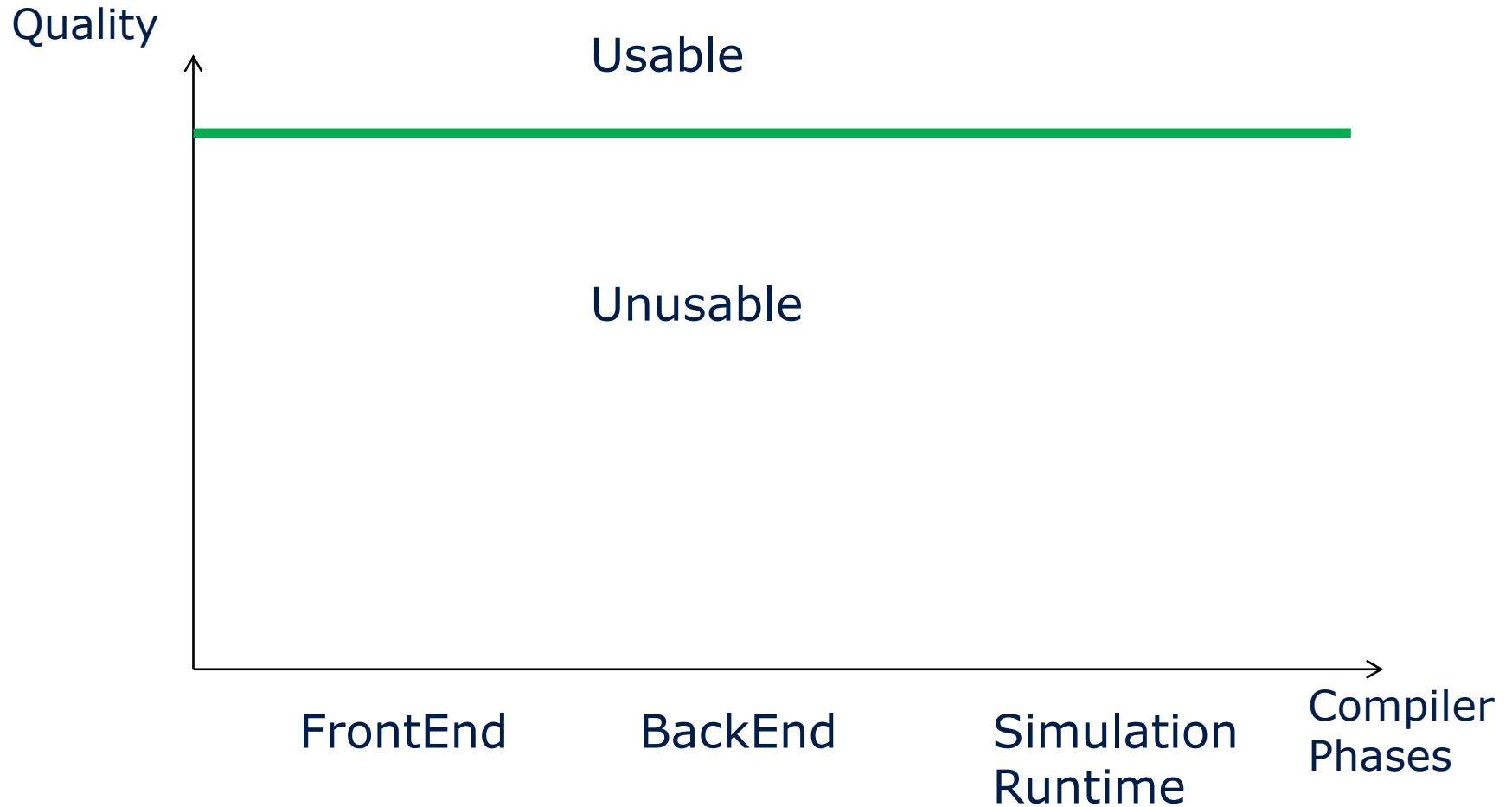
FrontEnd

BackEnd

Simulation
Runtime

Compiler
Phases

1. Introduction



1. Introduction

Quality

Target Groups:

- Modelica Developers
 - Compiler
 - Library
- Modelica Users
 - Commercial
 - Scientific
- Modelica Starters

OpenModelica

have to meet increasing requirements

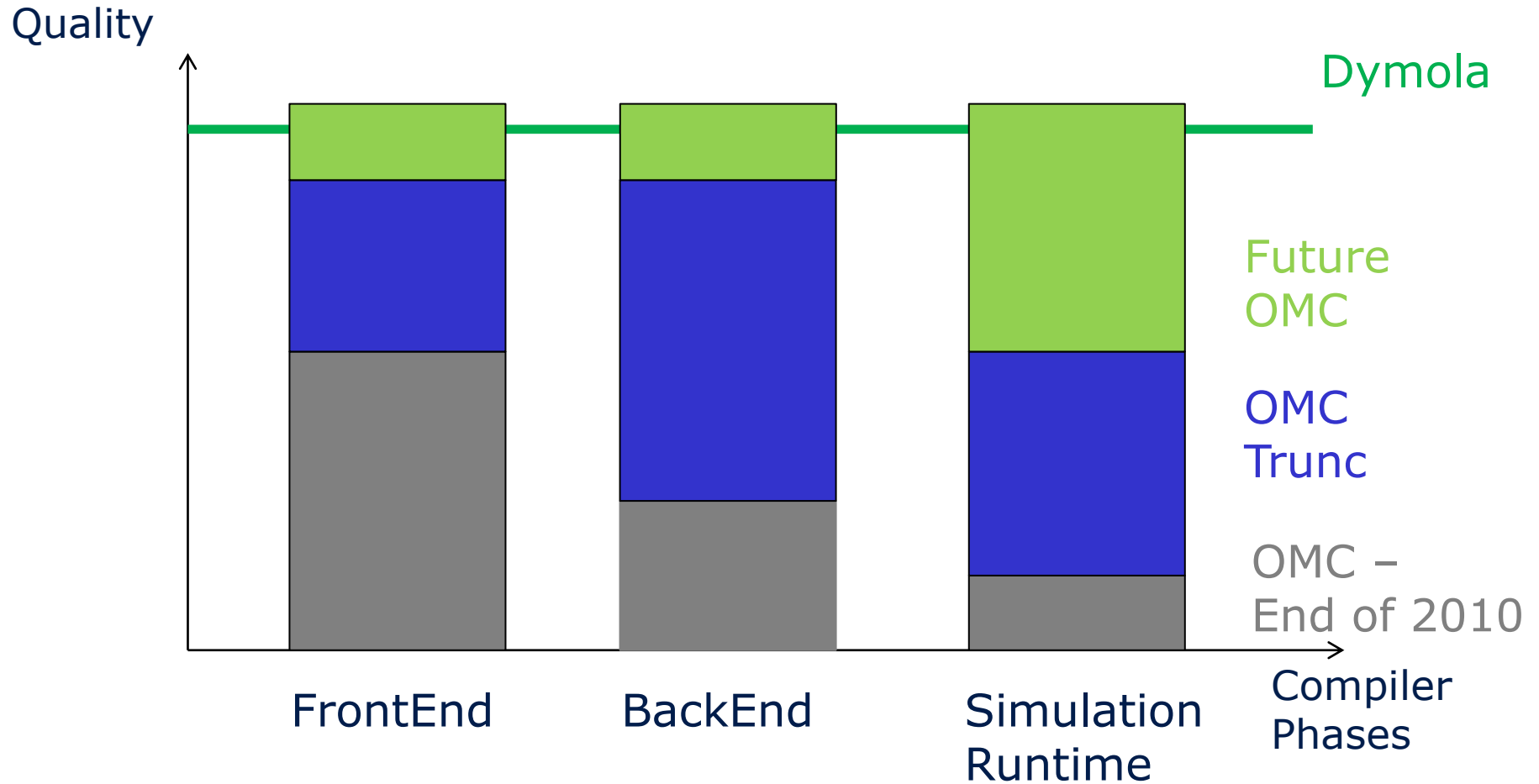
FrontEnd

BackEnd

Simulation
Runtime

Compiler
Phases

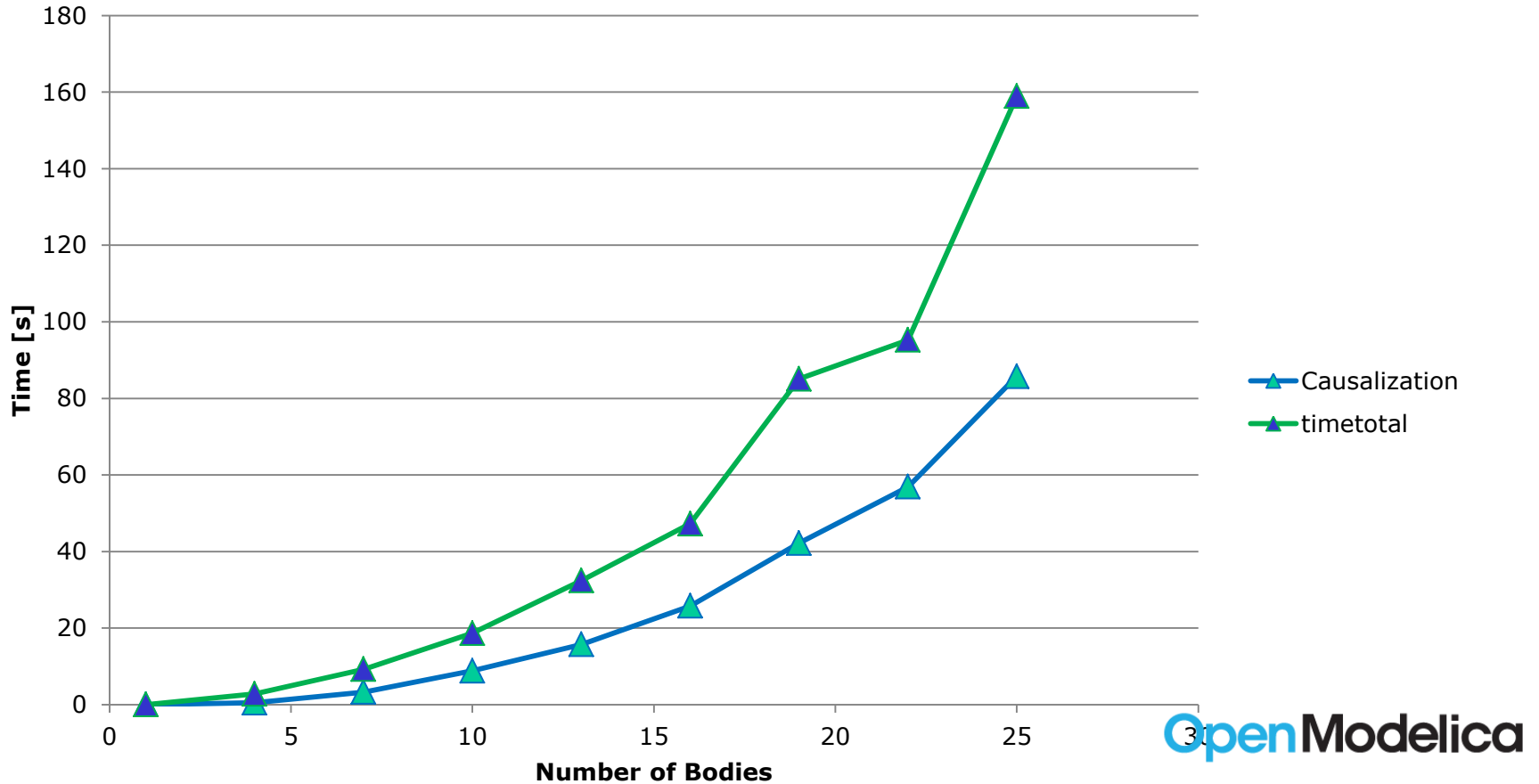
1. Introduction



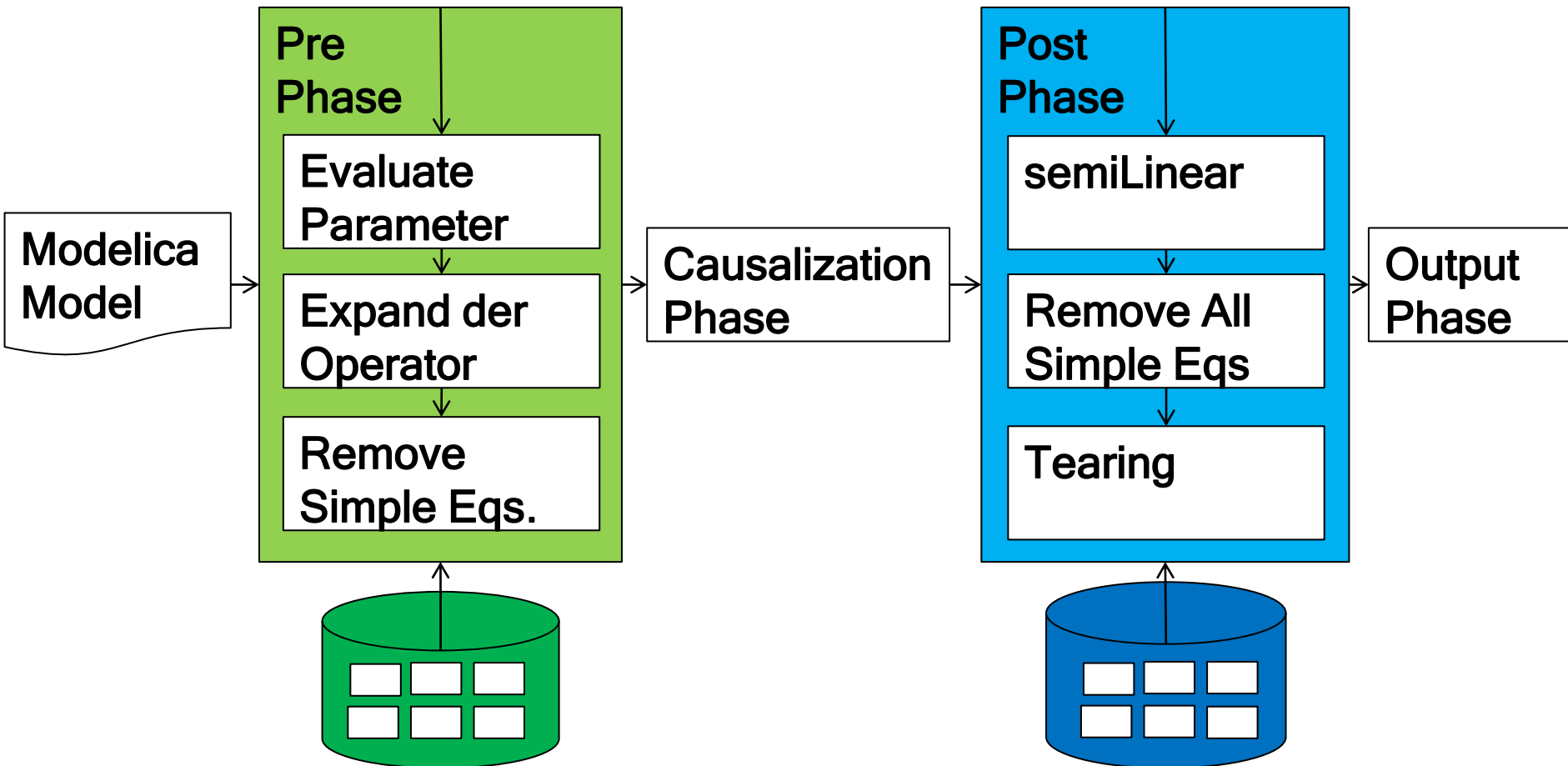
Problems - BackEnd - End of 2010

- **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?**

Pendulum - translation



Backend Pipeline



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**

Restructuring of Program Code

- **More fine grained package structure**
 - **More than 30 packages now**
- **Improve Maintainability**
- **Enhance Developing Process**

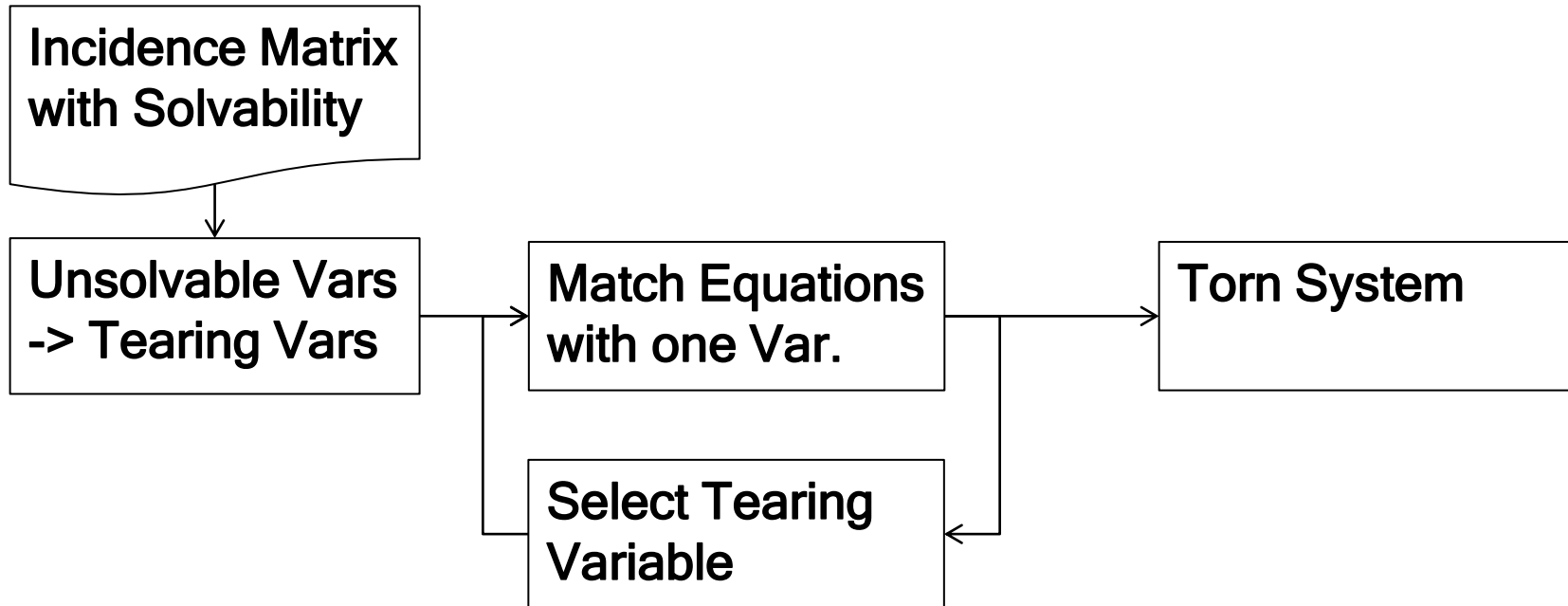
Support more Language Features in Data Structure

- **Complex Equations**
- **If-Equations**

Reimplement Support of Language Features

- **Array Equations**
- **Event Handling**

Tearing



Heuristic Based Selection:

- Unsolvable
- Wights on: Solvability, unassigned Edges, Type

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**

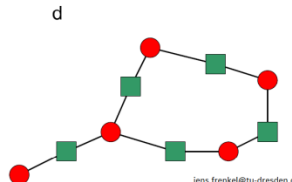
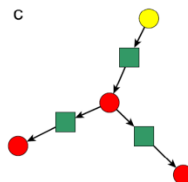
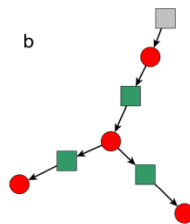
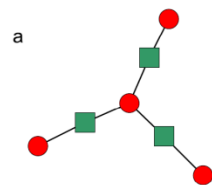
-> Volker Waurich (TU Dresden):

Comparison of Tearing Algorithms

Remove Simple Equations

- $a = b$
- $c = \text{constant}$
- **Almost linear scaling algorithm implemented**
- **Detect Singular Subsystems**

Collect Alias Sets



Select Alias Variable

Generate Replacement Rules

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,PFEExt,PFPLUSEExt,HKExt,HKDWExt,ABMPEExt,PREExt}
```

- **different heuristic based Matching Algorithms**
 - **Implemented in MetaModelica (2) and C (3)**
 - **speed up**

```
setCheapMatchingAlgorithm(3); {0,1,2,3}
```

- **Decrease Compilation Time**

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

Causalization

PrePhase

Singular System?

Index Reduction

Singular
Subsystems?

State Selection

BLT Form

PostPhase

- Usable with all Matching Algorithms
- allow higher derivatives, $\text{der}(x,3)$
 - no additional memory
- Consider State Order, $w=\text{der}(\text{phi})$
 - less Dummy Der. Variables
- Handle undifferentiable Subsets from `StateSelect.prefer` selection
 - `Modelica.Fluid.Examples.HeatingSystem`
 - `Modelica.Magnetic.FluxTubes.Examples.SaturatedInductor`
 - `Modelica.Magnetic.FluxTubes.Examples.SolenoidActuator.ComparisonPullInStroke`

Causalization

PrePhase

Singular System?

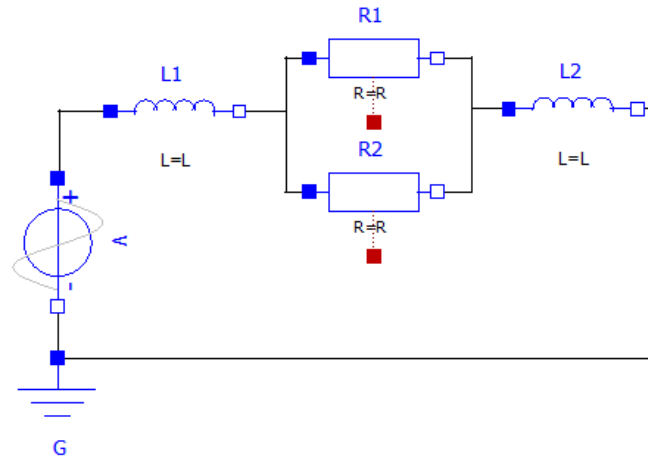
Index Reduction

Singular
Subsystems?

State Selection

BLT Form

PostPhase



$$\begin{aligned}
 R1.i + R2.i - L1.i &= 0.0 \\
 L2.i + (-R2.i) - R1.i &= 0.0 \\
 \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} * \begin{bmatrix} R1.i \\ R2.i \end{bmatrix} &= \begin{bmatrix} -L1.i \\ L2.i \end{bmatrix} \\
 \det((1*-1)-(-1*1)) &= 0 \\
 L2.i - L1.i &= 0.0
 \end{aligned}$$

Causalization

PrePhase



Singular System?



Index Reduction



Singular
Subsystems?



State Selection



BLT Form



PostPhase

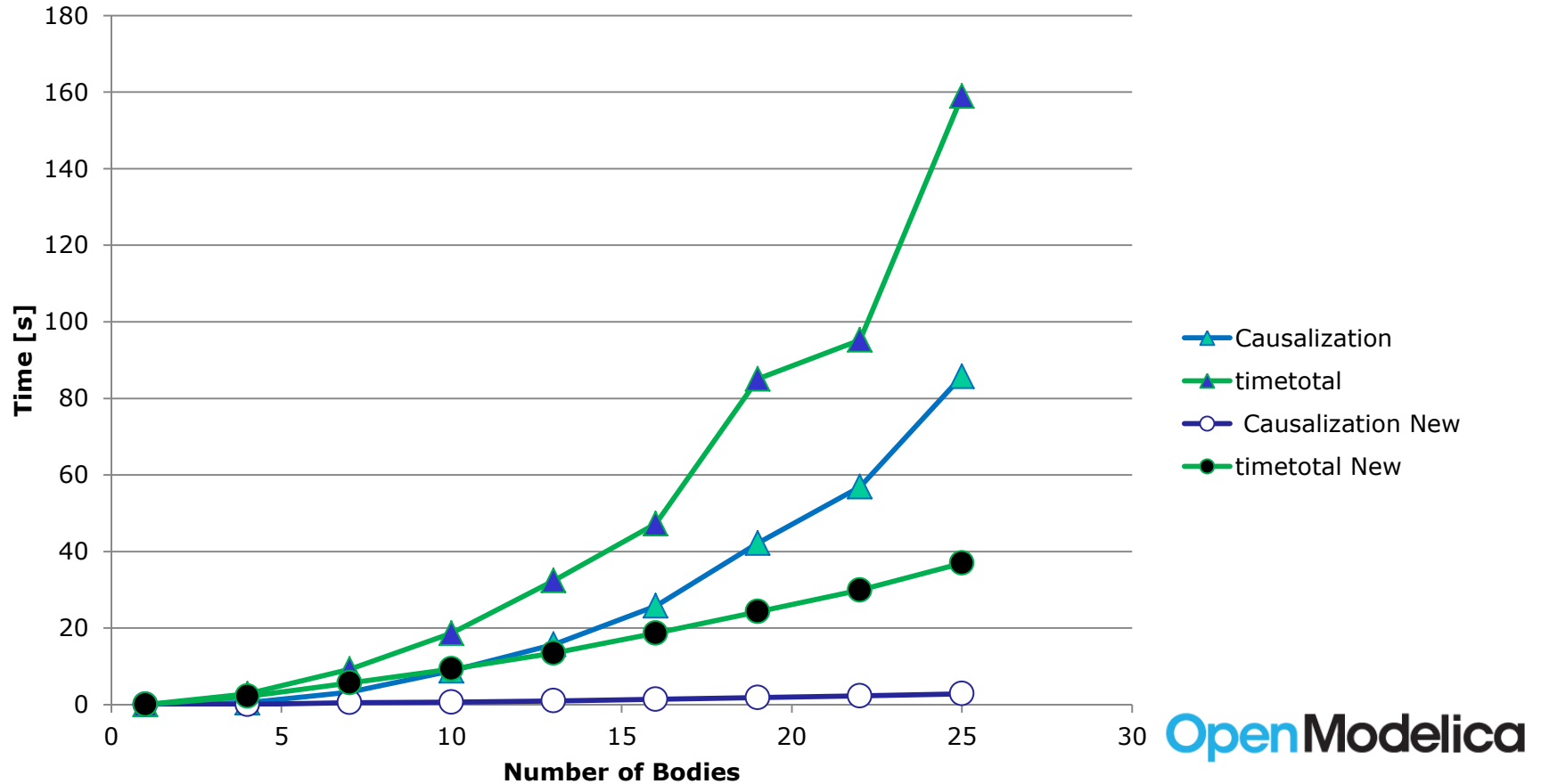
- **Based on Selection Algorithm by Mattsson and Söderlind**
- **Improved to consider StateSelection**
- **Avoid Algebraic Loops by Selection of States (heuristic)**

Symbolic Initialization

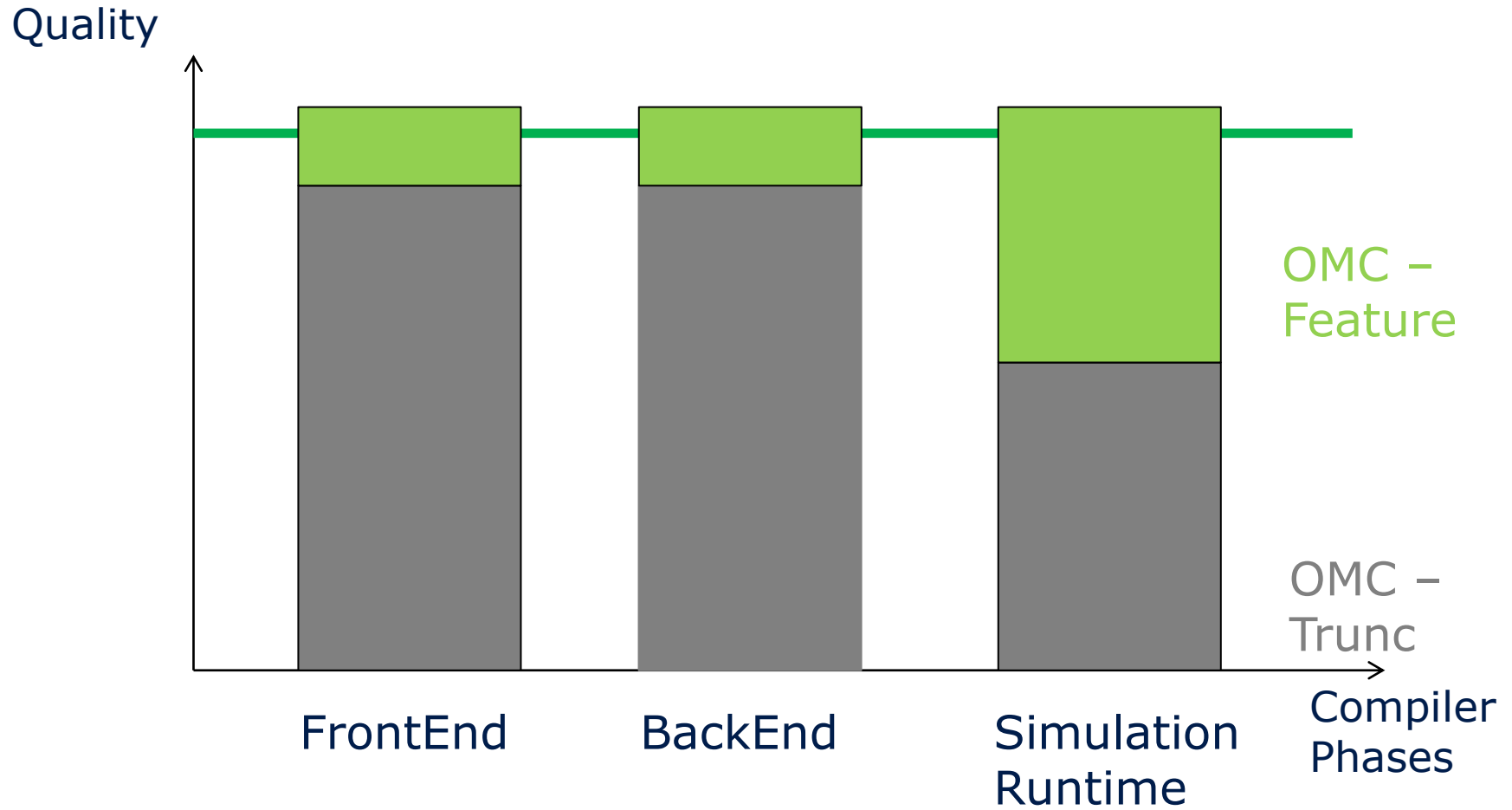
**Lennart Ochel, Bernhard Bachmann, Willi Braun
(FH Bielefeld):**

Initialization within OpenModelica

Pendulum - translation



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**



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