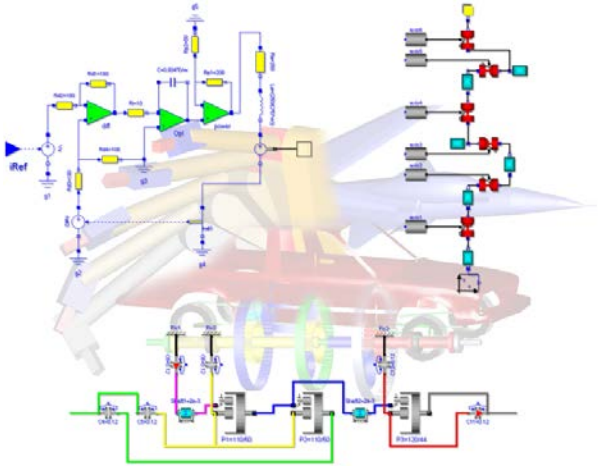
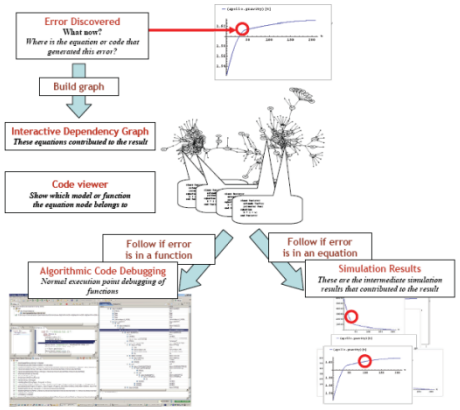
Design of the New OpenModelica Compiler High Performance Frontend

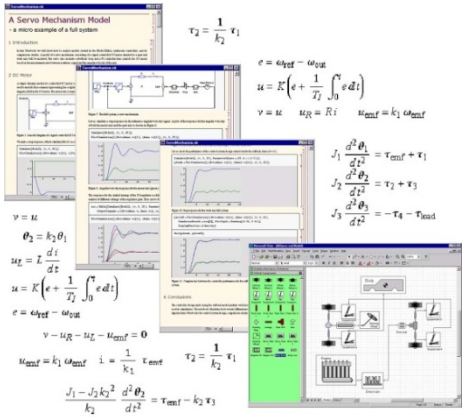
Per Östlund, Adrian Pop

2019-02-04

Open Source Modelica Consortium
Programming Environment Laboratory
Department of Computer and Information Science
Linköping University







www.OpenModelica.org

OPEN SOURCE MODELICA CONSORTIUM

OpenModelica

MODELICA

pelab

RI SE

LINKÖPING UNIVERSITY

Outline

- Motivation
- Architecture
- Design
- Benchmarks
- Conclusions

Motivation

- Current front-end (CF)
  - Makes it hard to fix issues
  - Is hard to extend with new features
  - Has issues with Scalability
  - Has issues with some Modelica support (#2858)
- Decision
  - Implement a new front-end (NF) from scratch

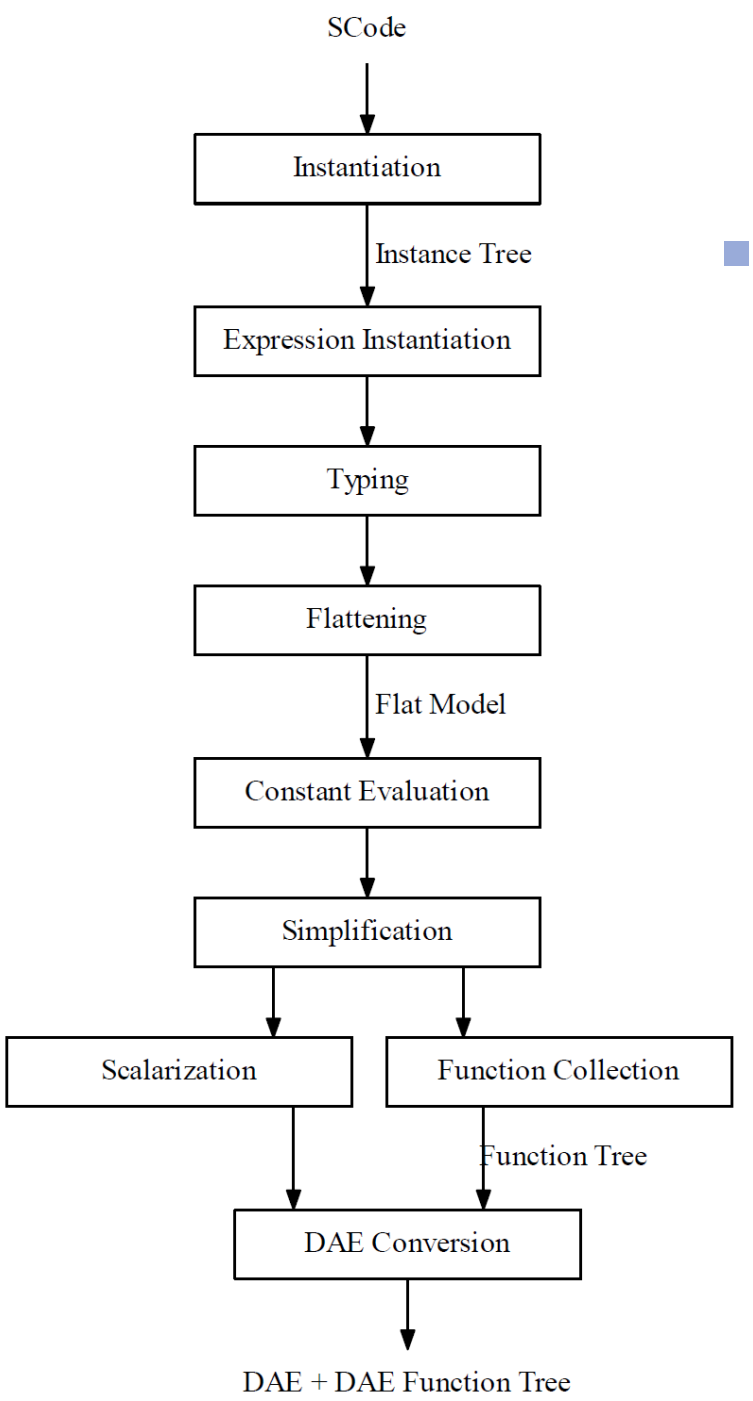
Architecture

- 65 Packages / Uniontypes
  - Written in MetaModelica 3.0
  - Much better modularization
  - Much easier to follow

```
encapsulated uniontype NFFlatModel
  import Equation = NFEquation;
  import Algorithm = NFAlgorithm;
  import Variable = NFVariable;

  record FLAT_MODEL
    list<Variable> variables;
    list<Equation> equations;
    list<Equation> initialEquations;
    list<Algorithm> algorithms;
    list<Algorithm> initialAlgorithms;
    Option<SCode.Comment> comment;
  end FLAT_MODEL;

end NFFlatModel;
```

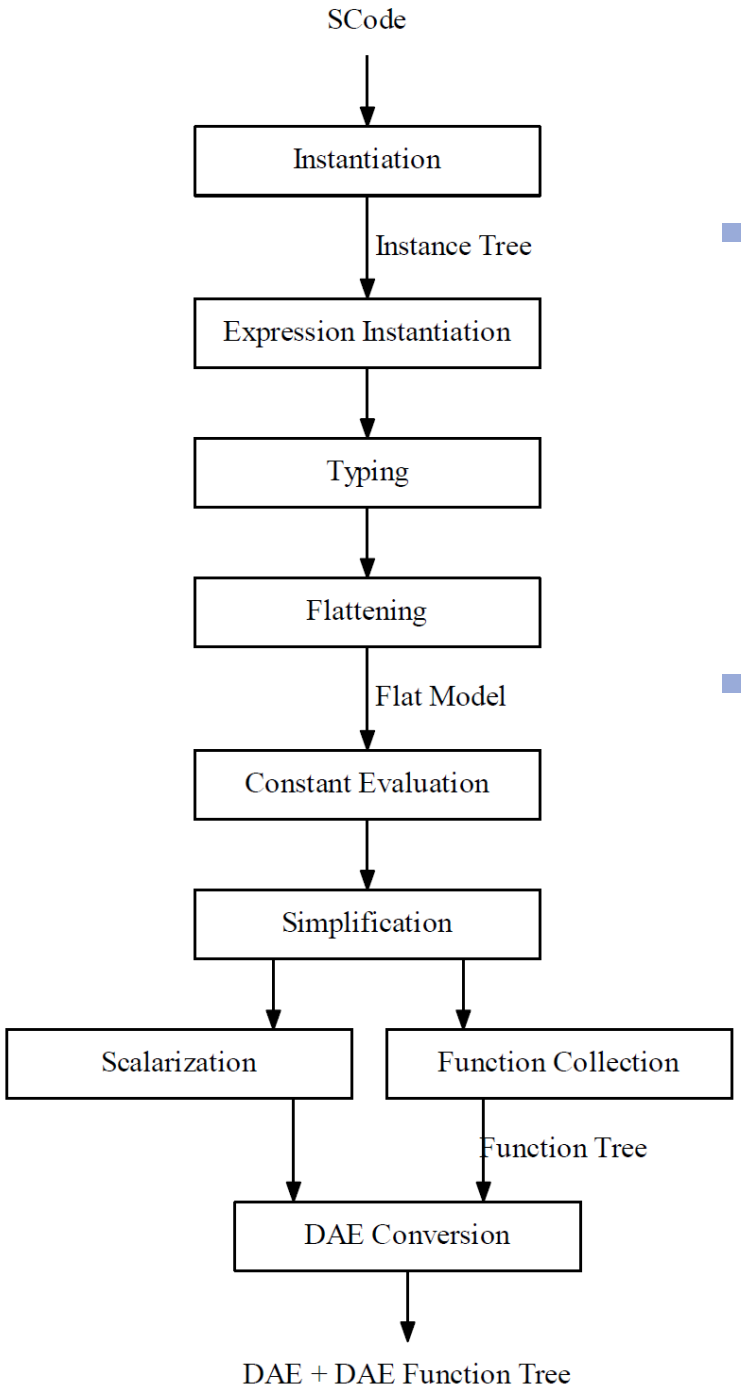
```
encapsulated package NFBinding
  public
    import Expression = NFExpression;
    import NFInstNode.InstNode;
    import SCode;
    import Type = NFType;
    import NFPrefixes.Variability;
    import Error;
  protected
    import Dump;
  public
    constant Binding EMPTY_BINDING
      = Binding.UNBOUND();

uniontype Binding
  record UNBOUND
  end UNBOUND;

  record UNTYPED_BINDING
    Expression bindingExp;
    // ...
  end UNTYPED_BINDING;

  record TYPED_BINDING
    Expression bindingExp;
    // ...
  end TYPED_BINDING;
```

```
public
  function isBound
    input Binding binding;
    output Boolean isBound;
  algorithm
    isBound := match binding
      case UNBOUND() then false;
      else true;
    end match;
  end isBound;

  function untypedExp
    input Binding binding;
    output Option<Expression> exp;
  algorithm
    exp := match binding
      case UNTYPED_BINDING()
        then SOME(binding.bindingExp);
      else NONE();
    end match;
  end untypedExp;

  function typedExp
    input Binding binding;
    output Option<Expression> exp;
  algorithm
    exp := match binding
      case TYPED_BINDING()
        then SOME(binding.bindingExp);
      else NONE();
    end match;
  end typedExp;

end Binding;
end NFBinding;
```



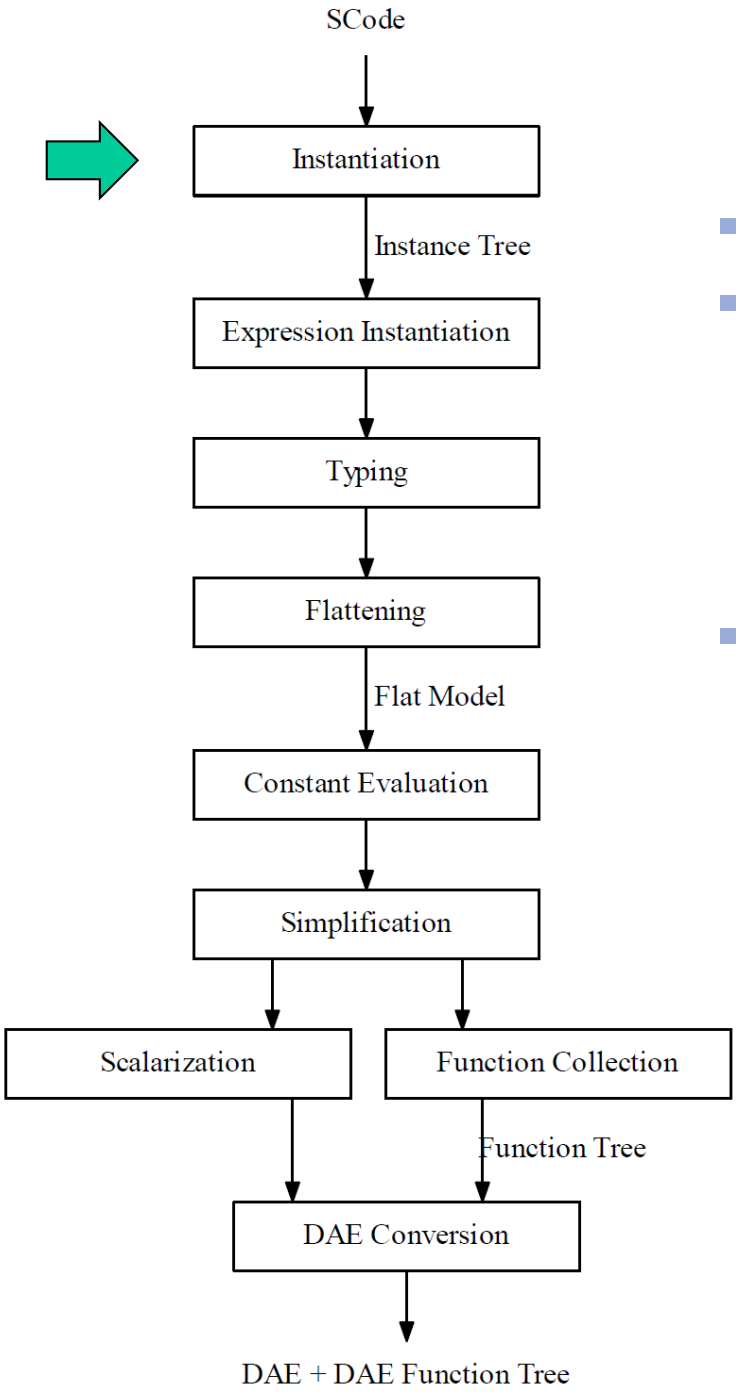
- Performance and Scalability
  - Pointers that link component references to definition and application (use)
  - Arrays of basic types and arrays of models are not expanded until Scalarization phase

Design – CF vs NF



- CF
  - Scode->DAE as an atomic operation for each component
  - Component focused
  - Dependencies are re-processed all over again
- NF
  - Find ways to break dependencies between the various frontend phases
  - Model focused (the entire model is processed before calling the next phase)

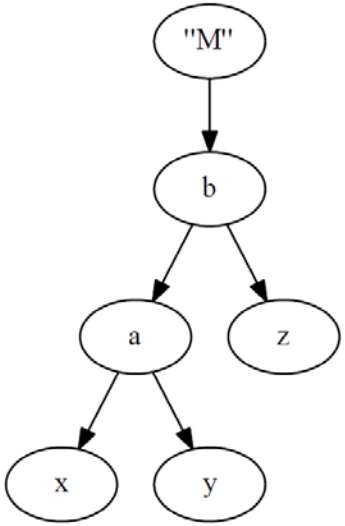
Design – Instantiation



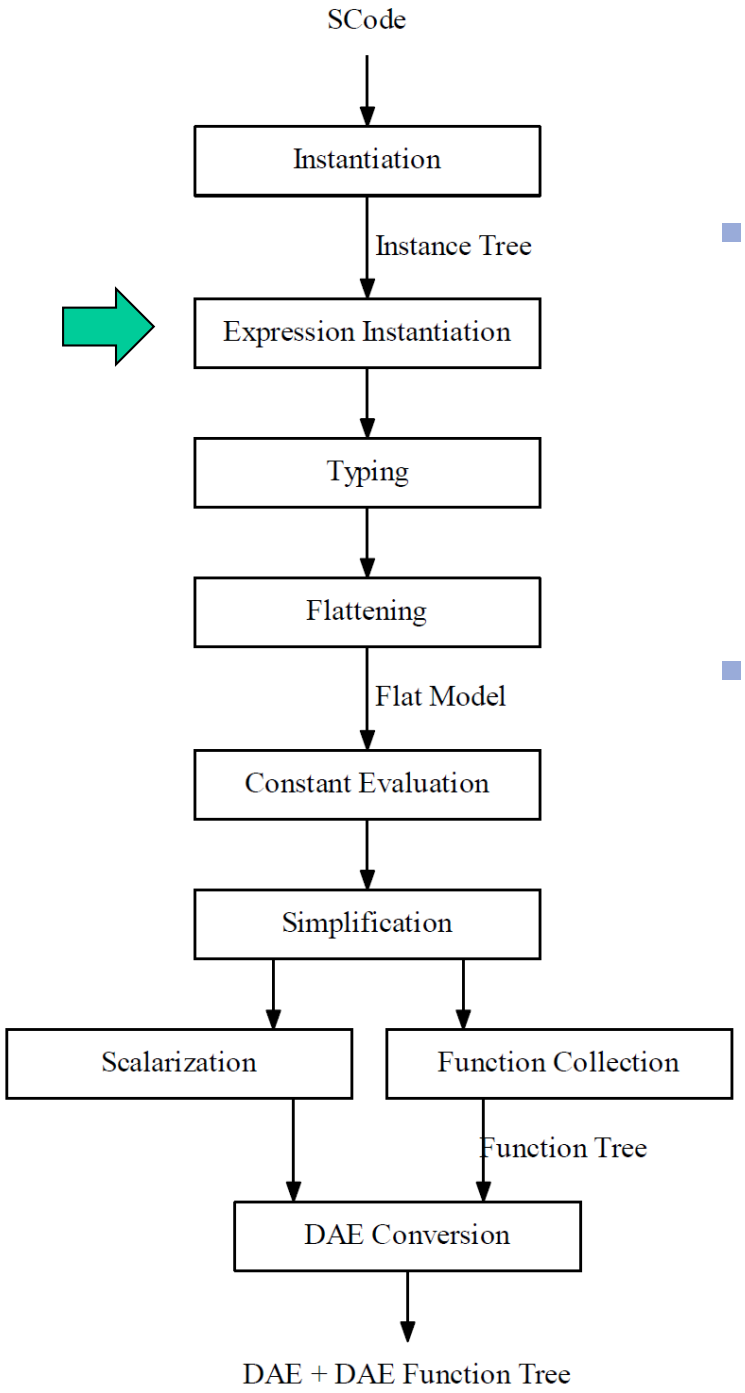
- Builds an instance tree for that model
- The instance tree consists of the class instance corresponding to the model as the root node, with the component instances of the class as child nodes that themselves have component instances
- Three stages
  - partial instantiation (local classes + imports)
  - expansion (base classes + local components)
  - full instantiation (modifiers)

```
model A
  Real x;
  Real y;
end A;

model B
  A a;
  Real z;
end B;

model M
  B b;
end M;
```

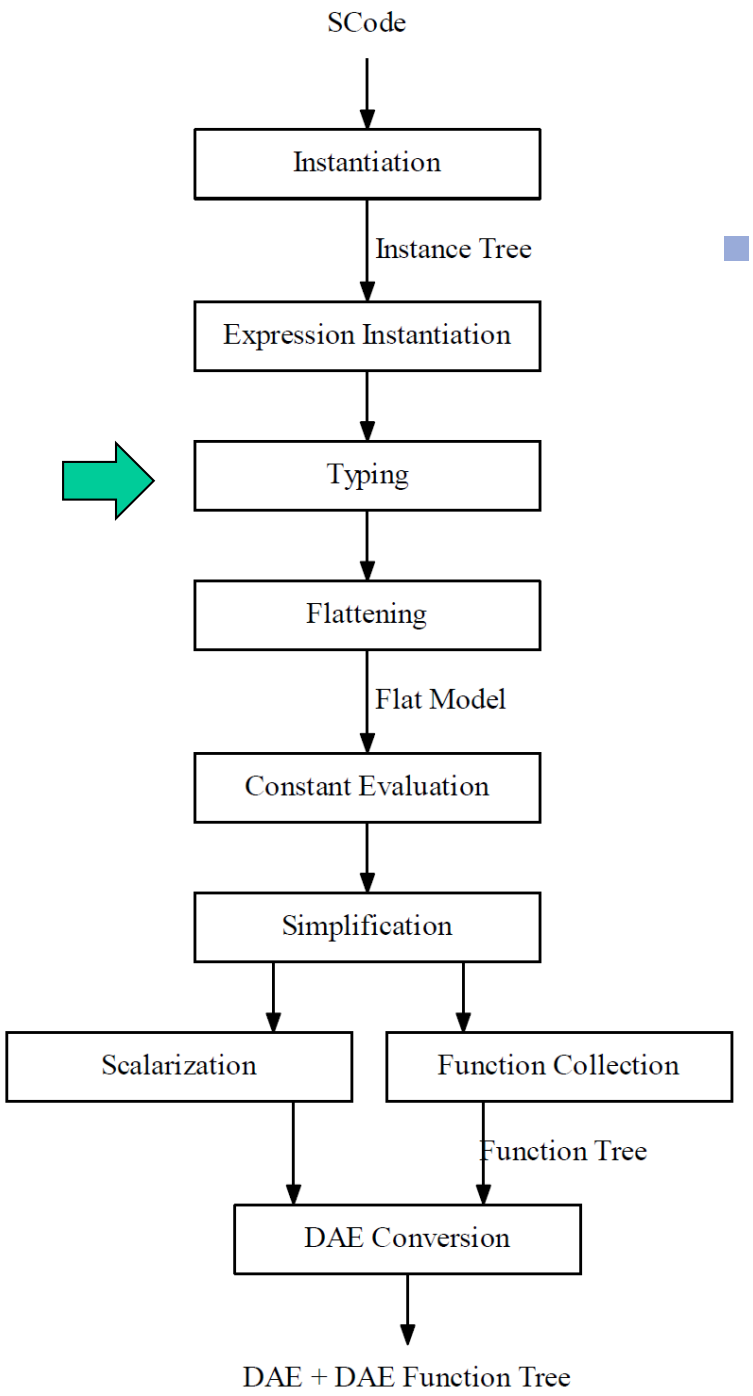
⇒



Design – Exp Instantiation



- CF
  - Component references are stored as (name, type, subscripts, next_component_reference)
  - Lookup needed again when extra info was required
- NF
  - Component references are stored as (reference_to_instance_tree, type, subscripts, next_component_reference)
  - Find correct instance via lookup
  - No lookup needed again (just once)

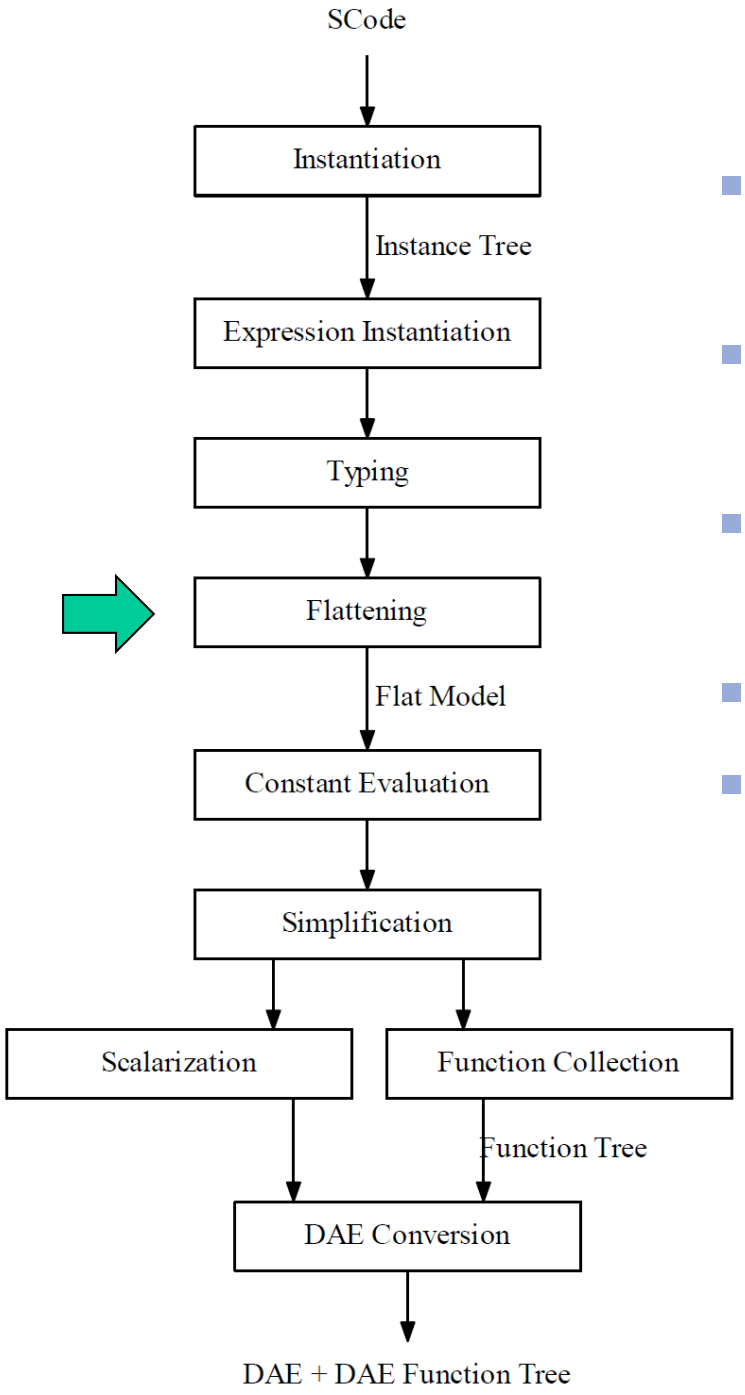


Design – Typing

- Two stages
  - Typing components, including dimensions
  - Typing expressions, bindings, equations, algorithms

```
model A
  Real x;
end A;

model M
  A a[3](x = {1, 2, 3});
end M;
```

Design – Flattening

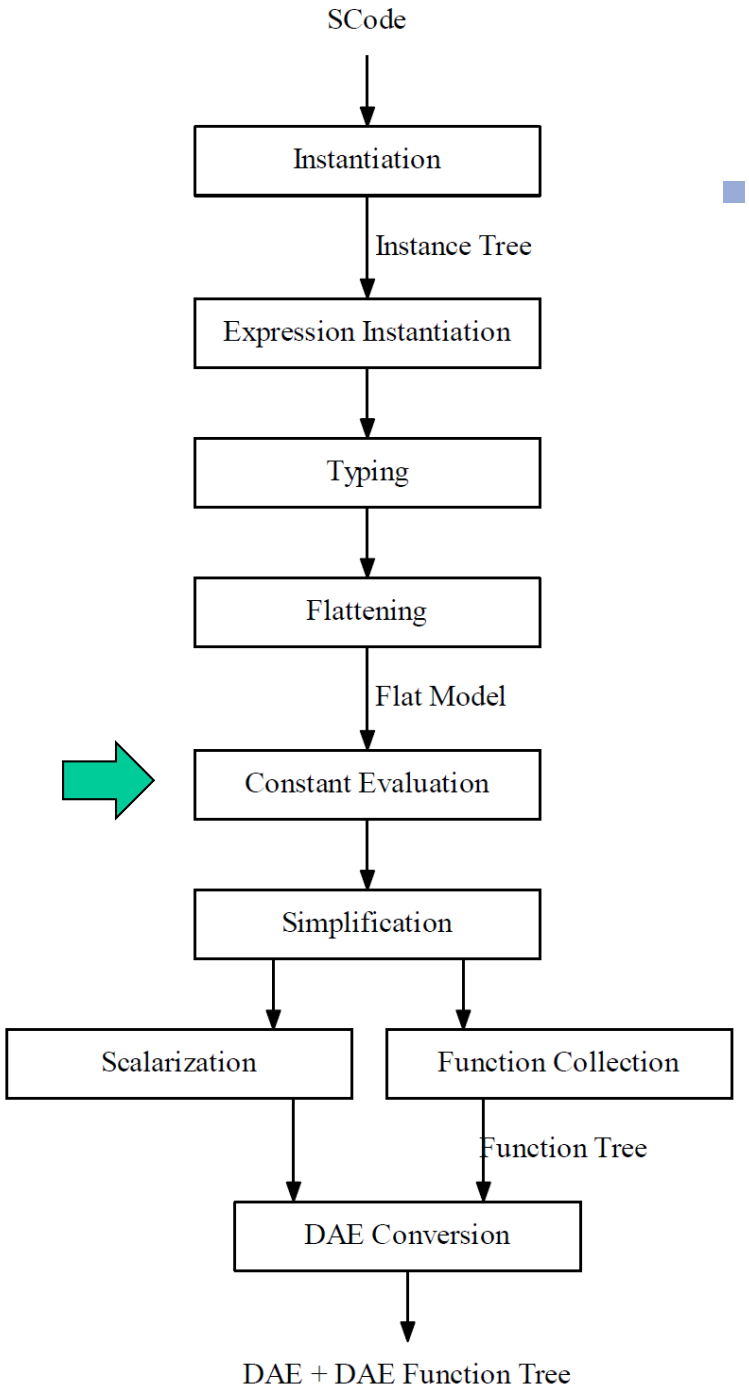


- Traverse instance tree and flatten it to a list
- Prefixing of component names and element name references
- Collect and handle connect equations, overconstrained connection graph
- Unroll for loops
- Scalarization of arrays of models

```
model A
  Real x;
  Real y;
equation
  y = der(x);
end A;

model B
  A a;
equation
  a.x = time;
end B;

model M
  B b;
end M;
```

⇒

```
model M
  Real b.a.x;
  Real b.a.y;
equation
  b.a.y = der(b.a.x);
  b.a.x = time;
end M;
```



Design – Constant Evaluation

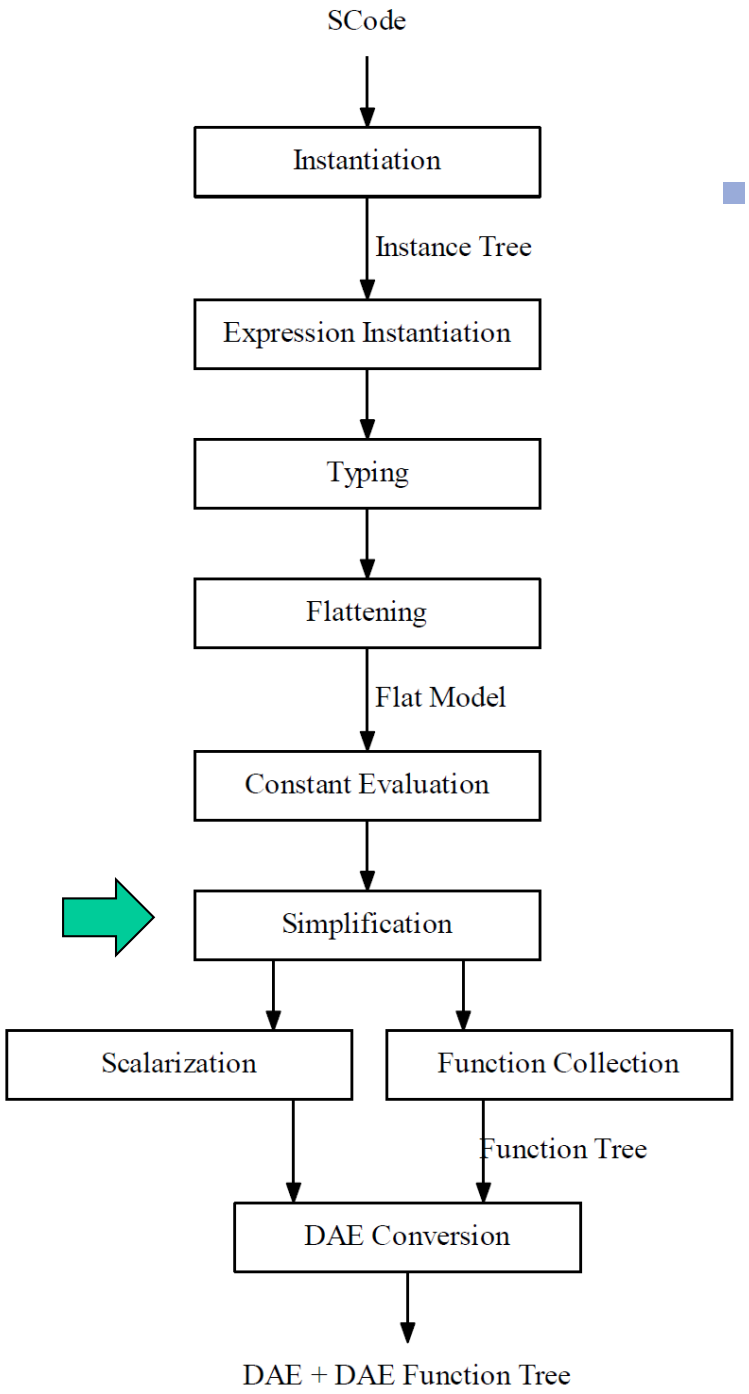
- Evaluate everything needed
  - Constants
  - Structural parameters (and Functions if needed)

```
model M
  constant Real x = 1.0;
  Real y;
equation
  y = x;
end M;
```

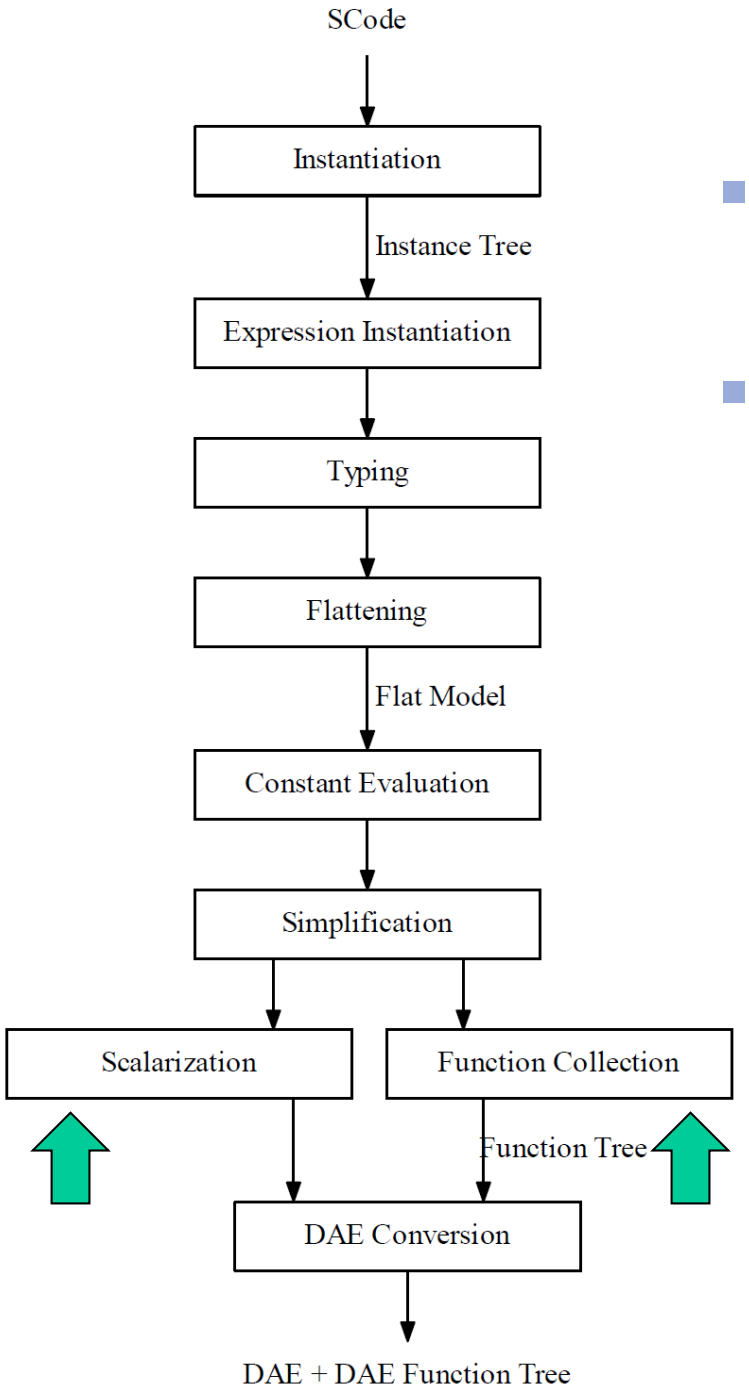
⇒

```
model M
  Real y;
equation
  y = 1.0;
end M;
```

Design – Simplification



- Simplify expressions, algorithms and equations
  - 1 + 2 => 3
  - Zero size for loops removed



Design – Scalarization

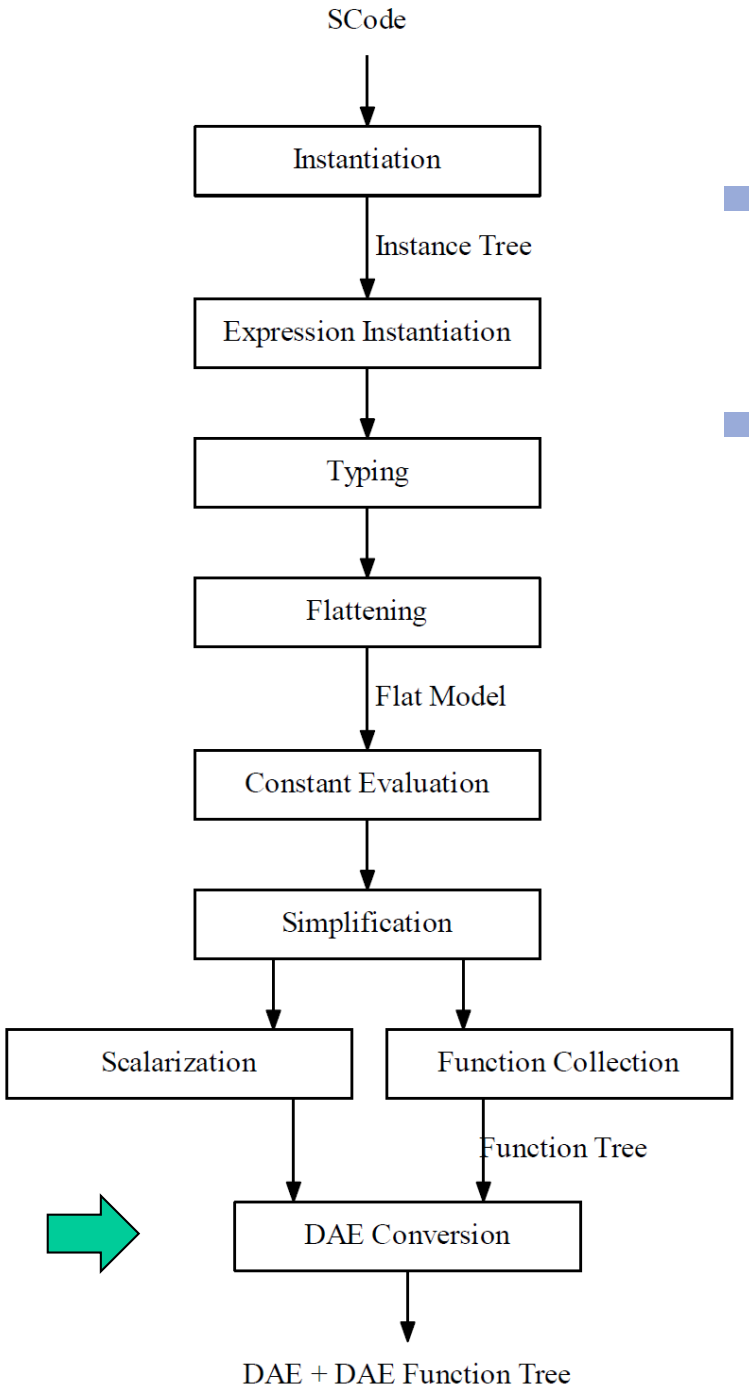
- Expand all arrays
  - Expand arrays of basic types
- Function Collection
  - Collect all functions used from expressions and derivative annotations

```
model M
  Real x[3];
equation
  x = {1, 2, 3};
end M;
```

⇒

```
model M
  Real x_1;
  Real x_2;
  Real x_3;
equation
  x_1 = 1;
  x_2 = 2;
  x_3 = 3;
end M;
```

Design – DAE Conversion



- Convert from FlatModel to DAE
- Convert all NF data types to DAE data types

Benchmarks

No	Model	Equations	Dym (s)	OMC NF/CF (s)
1	Electrical.DSystemAC.SE.DistributionSystemLinear_N_40_M_40	99776	15.53	06.32 / 91.33
2	Electrical.DSystemAC.SE.DistributionSystemLinear_N_80_M_80	397936	40.50	17.76 / 435.32
3	Electrical.DSystemAC.SE.DistributionSystemLinear_N_112_M_112	779312	74.21	32.31 / 1076.54
4	Electrical.DSystemDC.SE.DistributionSystemModelicaActiveLoads_N_80_M_80	129929	18.04	08.33 / 159.28
5	Electrical.TransmissionLine.SE.TransmissionLineModelica_N_1280	26915	09.84	04.45 / 47.77
6	Elementary.ParameterArrays.SE.Table_N_100_M_100	0	06.59	05.09 / 06.21
7	Elementary.ParameterArrays.SE.Table_N_400_M_400	0	10.25	12.19 / 18.03
8	Elementary.ParameterArrays.SE.Table_N_1600_M_100	0	09.77	19.04 / 28.17
9	Power.ConceptualPowerSystem.SE.PowerSystemStepLoad_N_64_M_16	11907	17.29	03.99 / 28.57
10	Vectorized.SolarSystem(n=10000) from section 4	60001	146.30	34.12 / 314.8 **(02.95)**
11	Vectorized.SolarSystem(n=100000) from section 4	600001	14458.68	2450.57 / 19760.42 **(02.95)**

Model	CF (s)	NF (s)	Factor
World	9.53	0.28	33.9
Joints.FreeMotionScalarInit	28.90	0.14	199.4
Joints.Planar	3.56	0.13	25.6
Joints.UniversalSpherical	6.99	0.22	30.5
Joints.SphericalSpherical	4.64	0.11	39.5
Joints.Universal	2.31	0.12	18.4

Conclusions

- NF – new frontend
  - Implemented from scratch
  - Designed for high performance and scalability
  - Speed comparable to commercial tools
  - Work in progress – MSL 3.2.3 423 of 424 models flatten

- Future work
  - Finalize the implementation (expandable connectors)
  - Test with more libraries and solve the remaining issues
  - Handle non-expanded array in the backend

Thank You!

Questions?

asodja, sjoelund.se, sebcoO11, lochel, wbraun, niklwors, hubert.thieriot, petar, perost, Frenkel TUD, Unknown, syeas460, adeas31, ppriv, riclis76, haklu, dietmarw, levsa, mahge930, xO5andfe, mohsen, nutaro, xO2lucpo, florosx, xO6hener, xO7simbj, stebr461, xO8joekl, xO8kimja, Dongliang Li, jhare950, x97davka, krsta, edgarlopez, hanke, henjo, wuzhu.chen, fbergero, harkaO11, tmtuomas, bjozac, AlexeyLebedev, xO6klasj, ankar, kajny, vasaie_p, niemisto, donida, hkiel, davbr, otto@mathcore.com, Kaie Kubjas, xO6krino, afshe, xO6mikbl, leonardo.laguna, petfr, dhedberg, g-karbe, xO6henma, abhinnk, azazi, xO2danhe, rrmusu, x98petro, mater, g-bjoza, xO2kajny, g-pavgr, xO5andre, vaden, jansilar, ericmeyers, xO5simel, andsa, leist, choeger, Ariel.Liebman, frisk, vaurich, mwalther, mtiller, ptauber, casella, vitalij, hkiel, jank, rfranke, mflehmig, crupp2, kbalzereit, marchartung, adrpo

OpenModelica Project

http://www.OpenModelica.org