Research Project Proposal:
High Performance Modelica Compilers

Marina Nikolic
marina.nikolic@mail.polimi.it
Computer Science and Engineering
Research Area(s)

• Theoretical Computer Science
  • Programming Language Theory
    • Compiler Construction
  • Modelling Languages

• Control and System Engineering
  • System Simulation
<table>
<thead>
<tr>
<th>JOURNAL TITLE</th>
<th>PUBLISHER</th>
<th>H-INDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM Transactions on Computer Systems</td>
<td>ACM</td>
<td>63</td>
</tr>
<tr>
<td>ACM Transactions on Programming Languages and Systems</td>
<td>ACM</td>
<td>63</td>
</tr>
<tr>
<td>ACM Transactions on Architecture and Code Optimization</td>
<td>ACM</td>
<td>32</td>
</tr>
<tr>
<td>Software – Practice and Experience</td>
<td>John Wiley&amp;Sons Inc.</td>
<td>63</td>
</tr>
<tr>
<td>SoftwareX</td>
<td>Elsevier BV</td>
<td>10</td>
</tr>
<tr>
<td>Advances in Engineering Software</td>
<td>Elsevier Ltd.</td>
<td>66</td>
</tr>
<tr>
<td>Simulation Modeling and Practice</td>
<td>Elsevier BV</td>
<td>58</td>
</tr>
<tr>
<td>Simulation</td>
<td>SAGE Pubblications</td>
<td>45</td>
</tr>
<tr>
<td>IEEE Transaction on Industrial Informatics</td>
<td>IEEE</td>
<td>100</td>
</tr>
</tbody>
</table>
Journals & Conferences (2)

- Compilers:
  - International Symposium on Code Generation and Optimization
  - International Conference on Compiler Construction
  - LLVM Developers’ Meeting

- Modelling and Simulation:
  - EOOLT Workshop
  - Modelica Conference
  - IEEE International Conference on Industrial Informatics
Modelica language is used for description and simulation of physical systems.

Properties:
- Non-proprietary
- Object-oriented
- Domain neutral
- Equation based
Modelica is used in both industry and academic research. The Modelica Standard Library contains about 1600 model components and 1350 functions from many domains.

Industries using Modelica are:

- Automotive industry, to design energy efficient vehicles or improved air conditioning systems (Audi, BMW, Daimler, Ford, Toyota, VW)
- Power plant providers (ABB, EDF, Siemens)
Compilation Steps

Resolution of the system of DAE
Transformation of the equation into code
Simulation and analysis of the results
Related works: tools and environments

- Commercial:
  - Dymola
  - AMESym
  - CyModelica
  - Wolfram SystemModeler
  - SimulationX
  - MapleSim
  - Catia

- Open source:
  - OpenModelica (www.openmodelica.org)
  - JModelica
Related works: partial projects (1)

Avoiding flattening and relying on LLVM:

Agosta, G., Baldino, E., Casella, F., Cherubin, S., Leva, A., and Terraneo, F.
Towards a high-performancemodelica compiler
Proceedings of the 13th International Modelica Conference, Regensburg, Germany, March 4–6,2019 (Jan 2019)
An alternative to Modelica tools:

Elmqvist, H., Neumayr, A., and Otter, M.
Modia - dynamic modeling and simulation with Julia
JuliaCon2018 (Jan 2018).
Related works: benchmark suites

• Frenkel J., Schubert C., Kunze G., Fritzson P., Sjölund M., and Pop A.  
  "Towards a benchmark suite for Modelica compilers: Large models"  
  *8th International Modelica Conference* (2011)

• Casella F.  
  "Simulation of large-scale models in Modelica: State of the art and future perspectives"  
  *11th International Modelica Conference* (2015)

• Agosta G., Casella F., Cherubin S., Leva A., and Terraneo F.  
  "Towards a benchmark suite for high performance Modelica compilers"  
  *EOOLT 2019*
Comparing compilation and execution time of Modelica tools against hand-written code, it can be seen how current to compilers’ performances are not satisfactory.
State-of-the-art compiler performances: space

- The image in this slide are from: Baldino E., Structural pitfalls of state-of-the-art modelica compilers: an explorative analysis. Master’s Thesis at Politecnico di Milano, 2018
Causes for long time-to-solution

1) Flattening

2) High space complexity - because of flattening
The goal is to contribute to the implementation of a Modelica compiler that generates code with **better performance** that state-of-the-art tools, by **preserving data structures** as much as possible (above all, **arrays**).

New **algorithms** have to be designed, and existing one to be adapted, in order to be able to exploit structural properties of the models.
Evaluation metrics

- Time complexity
- Space complexity

Both compilation and simulation will be evaluated, on new and existing benchmark suites.
Tasks

1. Preliminary work and state of the art
2. Benchmarking
3. Analysis of the state-of-the-art results, identification of the causes of the current performances
4. Design and implementation of algorithms that solve the problems analysed in task 3
5. Testing
6. Writing down the result and production of a conference paper