MASTER THESES IN DEVELOPMENT OF SIMULATION, OPTIMIZATION AND COMPILER TOOLS

Contents

Introduction

Simulation and optimization of dynamic systems is becoming a standard tool in several industrial branches. This trend is largely driven by the need to decrease product time-to-market. In order to increase productivity in the product design phase, model-based approaches are increasingly used. To meet the demand for model-based design methods and tools, domain specific languages have been developed. One such language is Modelica, which is a language targeted at modeling of complex heterogeneous physical systems. Modelica is currently used in a wide range of applications, including automotive systems, power plants, thermo-fluid systems, and robotics.

OPTIMICA Compiler Toolkit

Most of the theses proposals in this document are related to the OPTIMICA Compiler Toolkit (OCT) (https://www.modelon.com/products-services/modelon-creator-suite/optimica-compiler-toolkit/), which is an Modelica-based simulation and optimization environment, which also supports the Modelica language extension Optimica. A main part of the toolkit is the Modelica and Optimica compilers which are developed using the JastAdd framework. OPTIMICA Compiler Toolkit is maintained and developed by Modelon AB (http://www.modelon.com).

Contact: Christian Winther, Modelon AB

Runtime instantiation in OCT

Mostly, compilers perform runtime instantiation, allocating memory for instances during execution. In order to simplify the compilation process of Modelica the OCT compiler performs instantiation during early compilation. We have identified a few cases where it can be beneficial to delay the instantiation until execution, or at least code generation, which would lead to reduced execution time and memory consumption for the compiler.

The aim of this thesis would be to implement the delayed instantiation for a few cases and give us an understanding of how it would work in more complex cases.

Student profile: Two skilled and motivated students, who has taken a course in compiler construction (preferably with good grades), and has interest in compiler development and Java programming.

Contact: Jonathan Kämpe, Modelon AB

Interactive HTML diagnostics

Good compiler diagnostics is a powerful tool when debugging large models. Our simulation platform currently have limited debugging diagnostics in the form of a few static HTML pages that are generated during compilation. The goal of the proposed master's thesis project is to further improve compiler diagnostics reports and utilize the powers of HTML, JavaScript and CSS to offer the user with a more interactive diagnostics tool. The different parts of the project are:

Determining the requirements by surveying and interact with model developers and determine their workflow.

Improve existing HTML diagnostics by incorporating CSS and JavaScript to visualize the relationship between equations, variables and their computation order.

Student profile: One skilled and motivated student with interest in compilers, Java, HTML, JavaScript and CSS development, who has taken a course in compiler construction.

Contact: Johan Åkesson, Modelon AB

System modeling and simulation platform of the future

Modelon is building a system modeling and simulation platform of the future with the goal of creating a robust commercial platform available in the cloud. The solution needs highly interactive user interface for system modeling, massive parallelization of numerical simulations, post processing of large data sets and visualization in 2D and 3D. Collaborative system design preferred by modern engineers is a key capability of the platform, as well as high standards for IT security to keep sensitive customer product data safe.

As a master's thesis student, you will interact with experts within Modelon in a range of fields, including physical modeling, numerical algorithms, systems design and compiler technology, all disciplines needed to create a great system design platform. You will work closely with the development team and gain experiences with agile software development practices, including Scrum, code review and pair programming, all of which are key elements of Modelon's software development process.

We offer master's thesis projects in a **broad range of areas**, including:

3D visualization of system simulation, including vehicles

3D editing of system models

Customized client web apps to support design workflows in model-based design

Enhanced system modeling capabilities, e.g., icon editing and documentation generation, and composition of compiled models (FMUs)

Advanced and configurable visualization of computational results (3D plots, scatter plots etc.)

Student profile: One or two skilled and highly motivated student(s) with interest in numerical algorithms and UI programming. Prior knowledge of Python, JavaScript and 3D applications is considered a merit.

Contact: Johan Åkesson, Modelon AB