

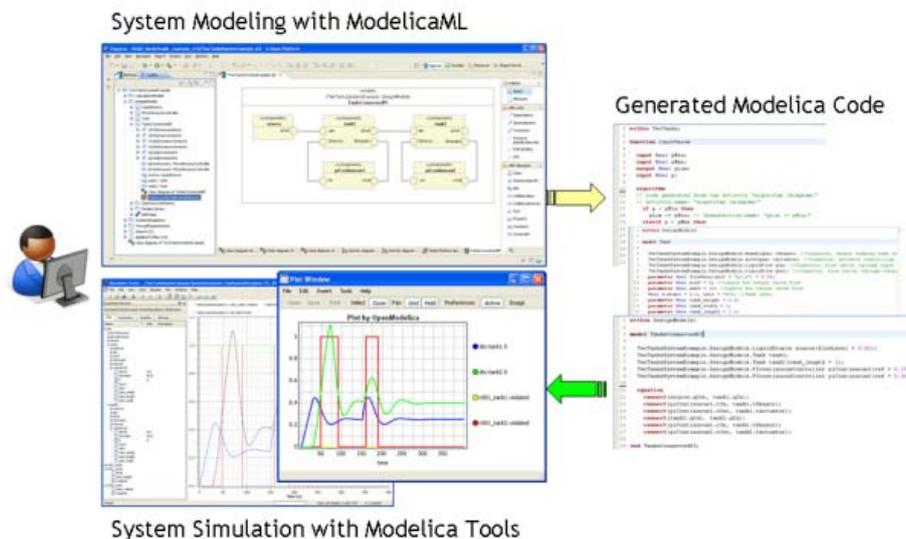
Master thesis

Support for the Authoring of Action Code in ModelicaML Models

Background - Why

One of the key enablers for model-based systems engineering is a consistent set of languages, graphical notations and their semantics, modelling methods and model transformations. ModelicaML is an attempt to answer to these needs by providing the engineers with an integrated system modelling environment that is based on standardized languages such as UML and Modelica.

Modelica Modeling Language (ModelicaML [5]) is a graphical modeling language for the description of time-continuous and time-discrete/event-based system dynamics. ModelicaML is defined as an extended subset of the [OMG Unified Modeling Language \(UML\)](#). This subset enables the generation of executable [Modelica](#) code. ModelicaML is an alternative way for authoring Modelica models. It extends the graphical modeling capabilities of Modelica by providing more diagrams (UML diagrams for presenting the composition, connection, inheritance or behavior of classes) for graphical model definition or documentation. Moreover, ModelicaML incorporates concepts for formalizing and evaluating system requirements during system simulations.



What

One of the challenges in ModelicaML is the handling of model-parts that are captured as strings (e.g. equations, statements, modifications that are stored as strings). The main task of this thesis is to develop and prototype an approach for capturing textual parts (e.g. equations, algorithm statements, etc.) of ModelicaML models in a way that is convenient to typical textual IDEs (e.g. JDK) but does not require refactoring of the textual parts of the model if the names of model elements change.

Therefore, make a literature review to find the “state of the art” in developing implementing domain specific languages using frameworks (e.g. XText[1], MPS[2]).

Focus should be on finding an approach to capture the information using a model in the background while still simulating a text-based editing of Modelica[3] code.

The most promising approach should be implemented as an Eclipse plug-in for Papyrus[4] or [6] and evaluated on test problems provided by EADS IW.

How

The Master thesis is done at Linköping University, Dept. for Computer Science and Information Science, in cooperation with EADS Innovation Works in Hamburg (Germany).

Supervisor: Dr. Adrian Pop, IDA, Linköping University, 58183 Linköping

E-mail: adrpo@ida.liu.se

Tel: +46 (0)13 285781

Other Contacts: Prof. Peter Fritzson, IDA, Linköping University, 58183 Linköping

E-mail: petfr@ida.liu.se

Contact person at EADS IW: Wladimir Schamai, EADS IW, Nesspriel 1, 21129

Hamburg, Germany

E-mail: Wladimir.Schamai@eads.net

Tel: +49 40 743 84 254

The Master thesis work is suitable for one or two students.

When: Now.

References

[1] Xtext is a framework for development of textual domain specific languages (DSLs, <http://www.eclipse.org/Xtext/>)

[2] JetBrains :: Meta Programming System, <http://www.jetbrains.com/mps/index.html>

[3]. Fritzson Peter, Principles of Object-Oriented Modeling and Simulation - with Modelica 2.1, 2004.

[4] Papyrus UML, <http://www.papyrusuml.org/>

[5] ModelicaML - A UML Profile for Modelica

<http://www.ida.liu.se/~pelab/modelica/OpenModelica/MDT/ModelicaML/>

[6] Papyrus UML, <http://www.eclipse.org/modeling/mdt/?project=papyrus>