

## Vertical Integration in Tool Chains for Modeling, Simulation and Optimization of Large-Scale Systems

J. Åkesson<sup>1</sup>,

<sup>1</sup>Modelon AB, Lund, Sweden and Lund University, Lund Sweden johan.akesson@modelon.com

In recent years, languages such as Modelica and VHDL-AMS have emerged as intuitive user and application-oriented high-level description formats suitable for modeling of physical systems. This trend has been further strengthened by the availability of software tools for modeling, simulation and optimization, which enable engineers to rapidly develop detailed models of complex systems composed from sub-systems from different physical domains. While the capabilities of such tools in terms of performance match the requirements in challenging industrial applications, tool interoperability has traditionally received little attention. Rather, tools have been designed as monolithic software environments, with dedicated interfaces to numerical algorithms. As a result, flexible creation of tool chains where several computational tools are assembled into a workflow tailored to a particular design process is often difficult. Driven by the observation that one single tool will not be able to provide the solution to the computational needs of the future, new challenges have emerged.

In order to meet these challenges, three aspects of computational tool chains need considered. Firstly, modeling languages are critical to provide comprehensive environments for engineering practitioners, as well as for enabling formal analysis of model properties. In addition, language extensibility, both in terms of the language itself and in terms of tools supporting it, requires attention in order to enable flexible tailoring of existing languages to specific needs arising when formulating different systems design problems. Secondly, open interfaces plays a key role in achieving tool interoperability. A recent example is the Functional Mock-up Interface (FMI), which has received considerable attention in the simulation tool community. Finally, symbolic and numerical algorithms designed to solve systems design problems need to be employed. In this presentation, challenges arising when integrating different tools into complete tools chains will be discussed. Particular attention will be given to the three mentioned aspects: languages, open interfaces and algorithms. Examples will be drawn from experiences from developing, integrating and using open source tools, notably CasADi and Jmodelica.org, in industrial projects where large-scale optimization techniques has been applied to Modelica models derived from first principles.