Ph.D. in Information Technology
Thesis Defense

June 3rd, 2026
at 2:00 pm
Room PT1 – building 20A

Michele SCUTTARI – XXXVIII Cycle

Advancing Modelica with MLIR: Array-Aware Compilation and Simulation

Supervisor: Prof. Giovanni Agosta

Abstract:

Modeling languages are becoming increasingly important in scientific and engineering domains, and they are driving the need for advanced compilers that automatically translate their descriptive semantics into efficient simulations. As models grow in complexity and scale, state-of-the-art compilers often encounter significant performance challenges, particularly when handling array-oriented models. These limitations hinder the adoption of modeling languages in high-performance applications and therefore restrict the expressiveness and flexibility available to domain experts.

This thesis addresses these challenges by presenting novel algorithms, data structures, and compiler infrastructure specifically designed for the compilation and simulation of large-scale array-oriented models written in the Modelica language. The work makes three principal contributions that collectively overcome the scalability limitations of existing compilation approaches.

The first contribution introduces compact data structures for representing arbitrarily high-dimensional index spaces, together with efficient algorithms for their manipulation during compilation. These structures maintain constant-time or logarithmic-time complexity for common operations when the number of scalar array elements – and thus equations – grows. Building upon these foundations, the thesis presents array-aware algorithms for incidence matrix construction, matching, strongly connected component detection, and equation scheduling that avoid the prohibitive costs of traditional scalar expansion approaches. These algorithms form the backbone of an array-aware causalization pipeline that processes models at the array level rather than decomposing them into individual scalar elements, whenever possible.

The second contribution addresses the need for modern compiler infrastructure capable of preserving high-level semantic information throughout the compilation process. The thesis presents a multi-level intermediate representation based on the MLIR framework, introducing custom dialects that capture differential and algebraic equation semantics, numerical solver interfaces, and runtime system operations. This modular architecture enables clear

separation of concerns, with each dialect addressing distinct aspects of the compilation pipeline while maintaining seamless integration with LLVM's mature optimization and code generation infrastructure.
The third contribution implements a multithreaded runtime environment engineered to exploit parallelism on modern multi-core processors. The compiler identifies independent equations and generates code that the runtime system distributes across available processor cores, translating compiler-identified parallelism into concrete performance improvements during simulation execution.
Experimental evaluation against the industry-grade OpenModelica compiler demonstrates the effectiveness of these contributions. The experimental results show that MARCO achieves near-constant compilation times as model size increases, contrasting sharply with traditional approaches that exhibit linear or worse scaling behavior. For instance, in the most complex examined case of object-oriented modeling of a silicon chip with cooling channels, MARCO compiled models with over 30 million scalar equations in approximately 2 seconds – the same time required for much smaller versions of the same model. In contrast, OpenModelica exhibited linear scaling, requiring up to ~30 minutes of compilation time for models with ~300 000 equations and being unable to handle models of larger size with the same hardware resources. Beyond compilation efficiency, simulation experiments demonstrated that the runtime environment efficiently exploits modern multi-core processors. As model size increases, MARCO's superior scalability becomes evident: for the same aforementioned thermal model, MARCO achieves simulation speedups ranging from 4.5x to 9x compared to OpenModelica, depending on the employed numerical integration method, but also up to two orders of magnitude for the simpler thermal version without cooling channels.
These results demonstrate that the combination of array-aware data structures and algorithms, modern multi-level compiler infrastructure, and parallel runtime support fundamentally transforms the scalability characteristics of differential and algebraic equation compilation. The presented solutions enable the practical simulation of large-scale cyber-physical systems that were previously computationally infeasible, paving the way for broader adoption of declarative modeling languages in high-performance computational science and engineering applications.

PhD Committee

Prof. Alessandro Barenghi, Politecnico di Milano

Prof. Biagio Cosenza, Università di Salerno

Prof. Giuseppe Tagliavini, Università di Bologna