The ChipsAct push to strengthen Europe's technological leadership brought new vital force to system level design, as a push to *"[…] build and reinforce [Europe's] capacity to innovate design, manufacturing, and packaging of advanced, energy-efficient and secure chips"*.

In this context, SystemC-AMS is a key resource, as flexible design and simulation framework that ensures flexibility to heterogeneous domains (discrete time, analog, mixed signal), and its adaptability to different abstraction levels, thus covering from system level down to digital and analog hardware descriptions and signal flow constructs.

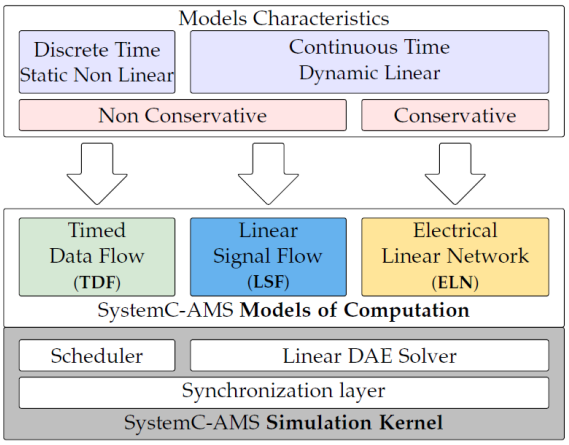

Figure 1SystemC-AMS organization and supported modeling styles

Despite of these strengths, SystemC-AMS still suffers severe limitation in the support of design:

- Lack of scheduling support: when a system is not schedulable, no information is provided to the user to help fixing the system;
- Limited tracing of signals (e.g., when signal flow values diverge);
- Lack of support for non linear components, that are nonetheless necessary to model a variety of components (e.g., diodes);
- Partial support of variable time step, that can be tuned by the user but is not automatically handled by the solver.

Additionally, SystemC-AMS effectiveness can be boosted by:

- Integrating additional solvers (like Runge-Kutta) in the kernel, to overcome the limitations of the standard one (Euler and trapezoidal methods), that fails at supporting dynamic systems;
- Analysis of parallelization alternatives for complex systems with a large number of components.

The proposes thesis will focus on a part of these topics, depending on the interest of the student and on current collaboration with companies that are active in the extension of SystemC-AMS and in the development of support tools.

Reading references:

[1] *Modeling and Simulation of Cyber-Physical Electrical Energy Systems With SystemC-AMS,* https://ieeexplore.ieee.org/document/8999635

[2] *An Introduction to Modeling Embedded Analog/Mixed-Signal Systems using SystemC AMS Extensions,* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.144.5749&rep=rep1&type=pdf

[3] *(Advanced material) SystemC and SystemC-AMS in Practice: SystemC 2.3, 2.2 and SystemC-AMS 1.0,* https://pico.polito.it/permalink/f/19j6qfa/TN_cdi_springer_books_10_1007_978_3_319_01147_9