Homogeneous code generation alternatives

**SystemC**
- Scheduling and data type support
- Generation of a simulatable implementation

**SystemC-AMS**
- Scheduling and data type support
- Support for continuous behaviors
- Generated code fully supports analog evolution

**Sequential C++**
- Simple semantics
- Approximates analog evolution
- Efficient execution and starting point of redesign flows

**Parallel C++**
- Support for parallel evolution of automata

**Massively parallel C**
- Support for parallel evolution
- Restrictions due to the architecture

Increasing complexity of embedded systems implies heterogeneity
- Different abstraction levels, digital/analog HW, embedded SW

Techniques for handling heterogeneity:
- **Top-down flows**
  - Model based design and formal models
  - Design flows with the definition of formal models
  - Restricted to certain application domains, heterogeneity is not handled
- **Bottom up flows**
  - Co-simulation connects frameworks and simulators specific to the different domains
  - No formal support, integration and validation is hard

Future work

**UNIVERCM**

**Static techniques:**
- Correct-by-construction manipulations:
  - RTL-to-TLM abstraction
  - Hardware-dependent Software generation
  - Model optimization
  - Formal verification
  - Based on Hybrid Automata Theory

**Simulation based techniques for Cyber-Physical Systems:**
- Functional requirements verification (e.g., dynamic Assertion Based Verification)
- Non-functional requirements estimation (e.g., power consumption)

**Homogeneous code generation**

**Water tank system: Code and model generation results**

A set of tools based on HIFSuite has been developed to generate both homogeneous code and high level models:
- Sequential C++
- SystemC
- SystemC-AMS
- SysML

Experimental results on a highly heterogeneous system, show advantages in producing homogeneous models:
- Faster simulation
- High level view of the system
- Easier integration of components
- Well defined integration of components

SysML model obtained starting from the water tank UNIVERCM specification: the structural description using a Block Diagram (top) and a Statechart Diagram representing the behavior of the valve component (bottom)

**References**

**Take Home Message**
Reducing Heterogeneity to Homogeneous models is the first step along the path to extend well known techniques for classical Embedded Systems to highly heterogeneous Cyber-Physical Systems. UNIVERCM features are perfectly suited to target this objective.