C based Modelling of Embedded Mixed Signal Systems

Institut für Computertechnik
ICT
Institute of Computer Technology

Ch. Grimm¹, K. Einwich²

¹TU Vienna
²Fraunhofer IIS/EAS Dresden
Motivations for C-Based approaches

- DSP methods include an increasing amount of control flow
  - SPUC, Ptolemy, …
  - HetSC, …

- DSP methods and AMS circuits are often functionally interwoven and can only be analyzed as a whole
  - MixSigC, ASC, SEAMS
  - SystemC-AMS
C based Modelling of Embedded Mixed Signal Systems

- Before (or without) SystemC
- SystemC based
- What’s next?
Very early approaches ...

- Many companies had in-house approaches, often starting in the 90s and maybe even earlier

- Motivation:
  - Simulator coupling backplane
  - Accelerated system simulation

- Example: AVSL [Meise et al.]
Signal Processing using C++ (SPUC, ’93-‘05)

- From TI/HP stuff???(Name of company has been removed …)
- Motivation
  - Faster System level simulation
  - Allows modelling of complex control flow in C/C++
- Open Source (sourceforge!)
- Most notably a DSP library in C/C++, no analog stuff
  - Modeling Digital Filter Functions, …
Ptolemy [Lee, Messerschmidt; ´90-´97]

- Used to study interaction of different simulators (MoC), Modelling/Simulation/Design of DSP systems
- Discrete-event, process networks, synchronous/reactive, and finite-state machine models of computation in C++
- Generates C-Code (or Assembler) for DSP realization
- Commercial Products: Agilent ADS, Cadence SPW
  \( \rightarrow \) Ptolemy II
Ptolemy [Lee, Messerschmidt; UCB]
Ptolemy II [Lee et al, 1996-now]

- Focus: Functional level modelling of DSP and other abstract heterogeneous systems
- Java based
- CT domain
  - DAE, interaction with DE and FSM domain
- Frequent use of polymorphism to allow use of components in different simulation scenarios
CT + „Mixed-Signal“ MoC in Ptolemy II ...

The sticky masses system has two modes of operation, "Separate" and "Together," corresponding to whether the point masses are stuck together. The "init" state has a transition that is used to initialize the "Separate" model (double click on that transition to see its actions).

Right click to look inside the states to see the model of the system behavior in the corresponding mode of operation.
"Wireless" MoC in Ptolemy II

This example shows a SoundSource (concentric circles icon) moving through a field of sensors (SoundSensor actors, with translucent circle icons) that detect the sound and communicate with a Triangulator actor (overlapping ellipses icon). The Triangulator performs sensor fusion to triangulate the location of the sound source, it generates a plot with estimated locations.

The SoundSource and Triangulator actors are composites, while the SoundSensor nodes are defined in Java. In all cases, you can look inside to view the implementation.

The sensors turn red when they detect a sound. Upon detecting a sound, they transmit the time at which they detect the sound and their current location.
VPPsim [Perrot, 2006]

Cadence Schematic Entry
- C++ primitive library
- Simulation parameter file

Automatic C++ system generation

C++ code

CppSim

Transient File Signals

Matlab processing scripts

Matlab
C based Modelling of Embedded Mixed Signal Systems

- **Before (or without) SystemC**

- **SystemC-based**
  - DE simulation kernel used
  - Extended kernel capabilities

- **What’s next?**
Behavioral modeling and simulation of mixed-signal system-on-a-chip using SystemC [Bonnerud,Ytterda2001]

- Purpose: Behavioural modelling of Mixed-Signal Systems
- No kernel extension
- Block diagrams with components for A/D Converter Design such as Sample & Hold, …
- No Netlists, transfer functions
- Use cases: Simulation of a A/D Converter
SystemC Kernel Extensions for Heterogeneous System Modeling [Patel, Shukla; 2002-2004]

- Purpose: Scientific work …
- Extends SystemC kernel capabilities

- SDF, FSM, CSP MoC integrated in SystemC-AMS
- MoC can be structured hierarchically

- Lacks support for CT domain

- … interesting: Comparison SDF with + without kernel extension (diophantine solver) →
SEAMS [T. Kazmierski et al, 2004]

- Purpose: Modelling + Simulation of AMS systems
- SystemC Environment with integrated AMS solver
- Network elements, Nodes
- http://eprints.ecs.soton.ac.uk/9413/
SystemC-WMS [Orcioni, Conti, 2003-now]

- Hard to classify ... distributed solver, mostly DE
- Two-ports connected via wavecannels solve equation systems in DE simulator.
- [http://www.deit.univpm.it/systemc-wms/](http://www.deit.univpm.it/systemc-wms/)
HetSC [Herrera, Villar; 2003 - now]

- Application:
  - HW/SW Codesign, SW synthesis
  - Class library of MoCs based on the DE MoC

- DE MoC implements
  - SDF, SR, DDF, … MoC

- Channels can convert MoCs where necessary
BeCom [Meise, Grimm, Waldschmidt]

- started 2001-2006, 2007-
- Symbolic preprocessing with MAPLE
  - reads SABER netlist
  - linearized equations in different workpoints
  - symbolically solves equation system
  - generates C-Code that
    - Computes output iff output signal is read
    - Computes states iff workpoint is left
  - Really fast …
ASC library [Grimm, Waldschmidt; 2001-2004]

- Analog/Mixed-Signal SystemC
  - Block diagrams, transfer functions, step width control
    + refinement of implicit control to control signals
  - Manual scheduling or 1 delay/block …
- Focus:
  - Investigation of Refinement Methodology for AMS and multi MoC systems

- OO-Design
  - Interface Classes allow designer to change MoC
  - Signals in this case adapt automatically (Polymorphic signals)
MixSigC [Einwich et al., 2000-2003]

- Cooperation Infineon + FhG-IIS/EAS Dresden
- SDF Engine + Linear Solver
- Mostly C-Based, no OO Design
- 2003 re-implementation with improved OO design, interface classes as evaluated in ASC
- ~2003 initial SystemC-AMS SG prototype
- 2005 submitted to OSCI
SystemC-AMS SG prototype

- Joint effort by SystemC-AMS SG
  - Base on re-implementation of MixSigC
  - New, innovative OO design with some features from ASC
- Starting point for SystemC-AMS WG effort
C based Modelling of Embedded Mixed Signal Systems

- Before or without SystemC
- SystemC-Based
- What’s next?
Open issues (besides standardization …)

EMS Application

Function

Interaction
TLM / SystemC-AMS

Implementation

- HW/SW
- AMS/RF
- Matlab/Simulink
- Ptolemy II, …
- SystemC Functional
- SystemC
- TLM, VAST, …
- Verilog
- VHDL
- Verilog-AMS
- VHDL-AMS
- SystemC-AMS