



C based Modelling of Embedded Mixed Signal Systems

A decorative graphic consisting of a horizontal bar with a gradient from dark blue to light grey, and a small cluster of blue and grey squares on the left side.

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
- → Ptolemy II



Ptolemy [Lee, Messerschmidt; UCB]

The screenshot displays the Ptolemy II simulation environment with several components:

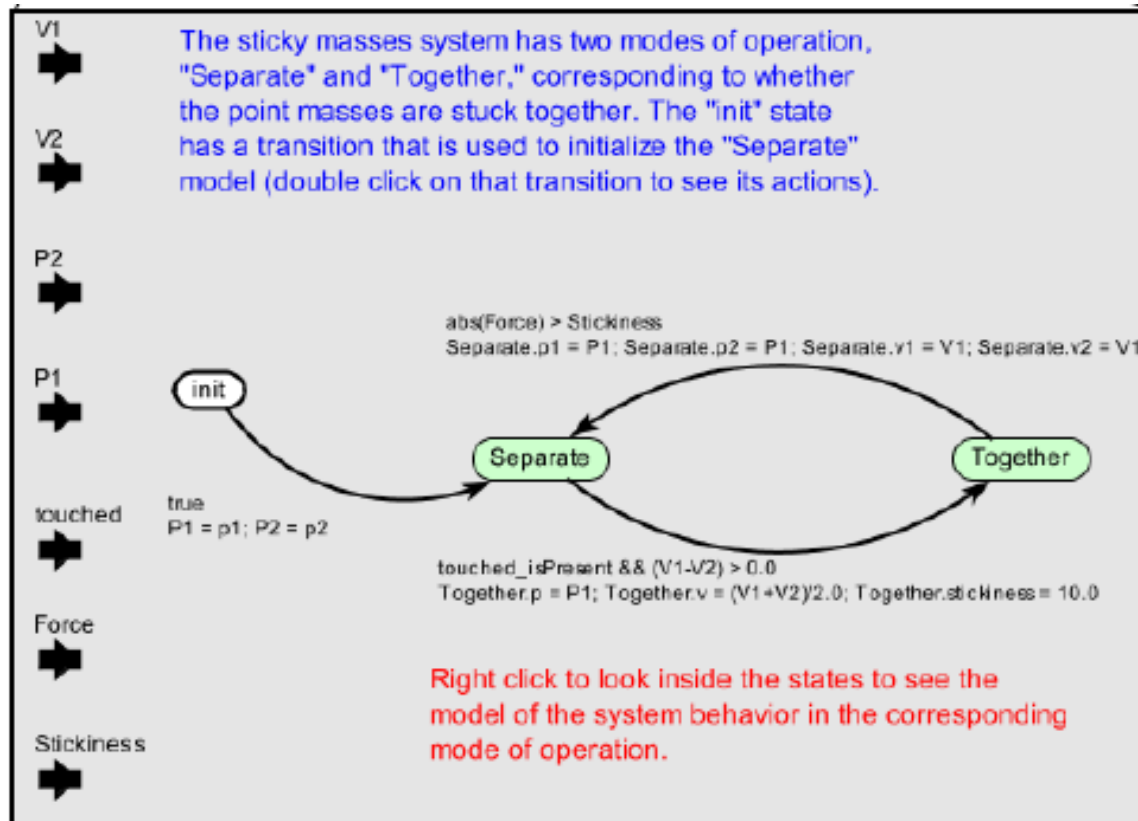
- Top Left Schematic:** A block diagram showing two 'Ramp' blocks connected to an 'Add' block, which is then connected to an 'Mpy' (multiply) block. A 'D.C.' block is also connected to the 'Mpy' block. The simulation window shows coordinates [-125, 125] and [963, 125].
- Top Right Thor Analyzer:** A window titled 'Thor Analyzer' showing the 'Ptolemy/Thor analyzer (analyzer.inputs=51)' with a 'simulator status: running'. It displays a 'CLOCK1.clk' signal as a square wave and several digital signals (DSP5600C1.RD, DSP5600C1.WR, DSP5600C2.WR, DSP5600C2.RD) as step functions. The time axis ranges from 0 to 200.
- Bottom Left DSP560002 Simulator:** A window titled 'DSP560002' showing the 'MOTOROLA DSP56000 SIMULATOR'. It displays a command window with the following text:


```
command1.cmd
RPC connection with Thor established
load code1
Loading file:code1.tod
go
SIMULATION ABORTED
x= $00000000000000 y=
a= $000000001000000 b=
x1= $0000000 x0=
y1= $0000000 y0=
a2= $00 a1= $0000001 a0=
b2= $00 b1= $0000000 b0=
pc= $0011 sr= $0310 omr=
la= $0000 lc= $0000
ssh= $0000 ssl= $0000 sp=
cyc= $000068 ictr= $000012 cnt1
P: $0010 0c0001 - JMP <
go
command0.
RPC connection with Thor established
load code0
Loading file:code0.tod
go
SIMULATION IN PROGRESS. Enter Ctrl-C to Halt. dev:0 pc:0011 cyc:68
go [(from)location/(reset)] [(to break number)/bn] [(occurrence)/count]
```
- Bottom Right C656MultiSim_2 Schematic:** A window titled 'C656MultiSim_2:schematic' showing a complex circuit diagram. It features two 'DSP56000' blocks (highlighted in yellow) connected to a central 'C656' block (highlighted in orange). The circuit includes various power supply connections (Vcc, GND) and signal lines. The simulation window shows coordinates [0, 0] and [1250, 225].

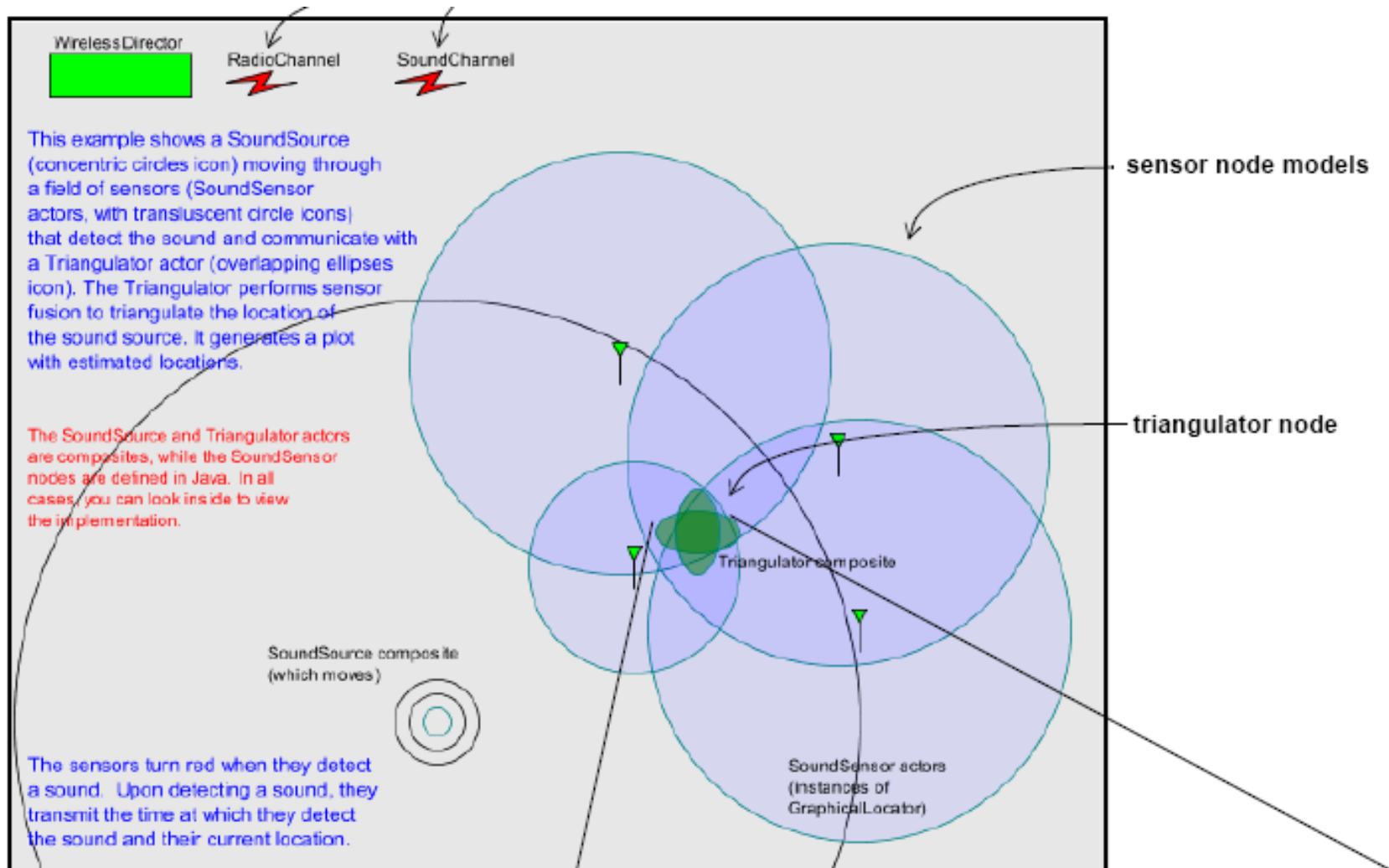
Ptolemy II [Lee et al, 1996-now]

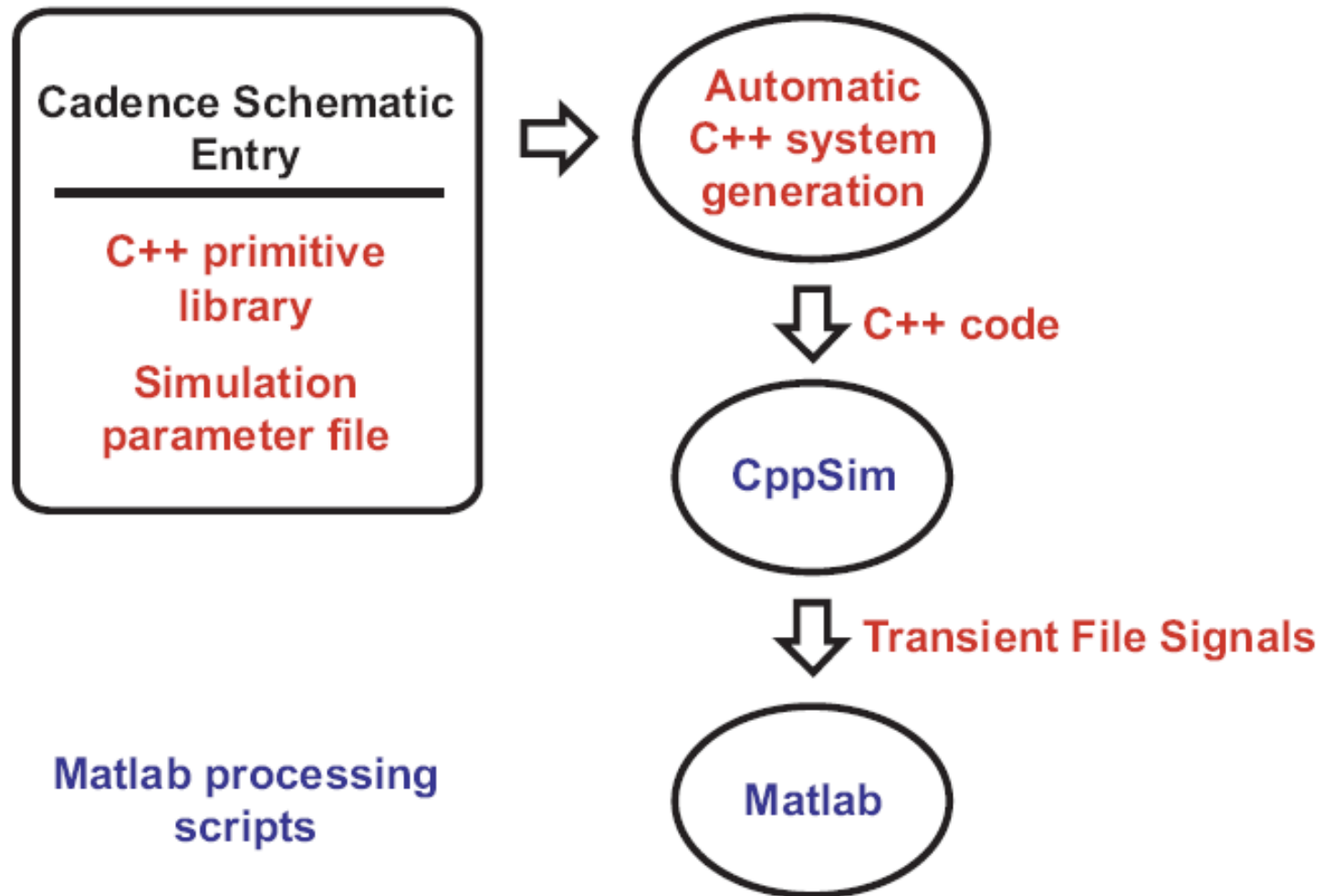
- **Focus: Functional level modelling of DSP and other abstract heterogeneous systems**
- Java based
- CT domain
 - DAE, interaction with DE and FSM domain
- Frequent use of polymorphism to allow use of components in different simulation scenarios

CT + „Mixed-Signal“ MoC in Ptolemy II ...



„Wireless“ MoC in Ptolemy II





C based Modelling of Embedded Mixed Signal Systems

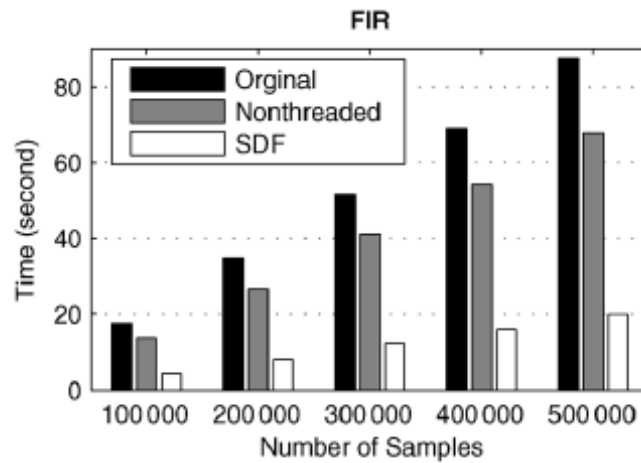
- **Before (or without) SystemC**
- **SystemC-based**
 - DE simulation kernel used
 - Extended kernel capabilities
- What's next?

Behavioral modeling and simulation of mixed-signal system-on-a-chip using SystemC [Bonnerud,Ytterda2001]

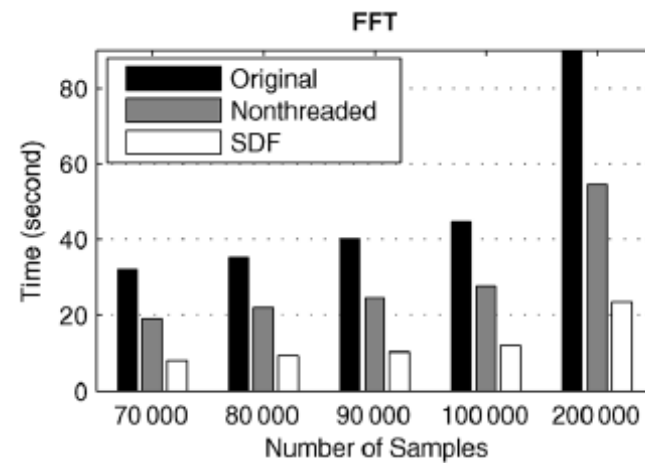
- Purpose:
Behavioural modelling of Mixed-Signal Systems
- No kernel extension
- Block diagrams with components for A/D Converter
Design such as Sample & Hold, ...
- No Netlists, transfer functions
- Use cases: Simulation of a A/D Converter

SystemC Kernel Extensions for Heterogeneous System Modeling [Patel, Shukla; 2002-2004]

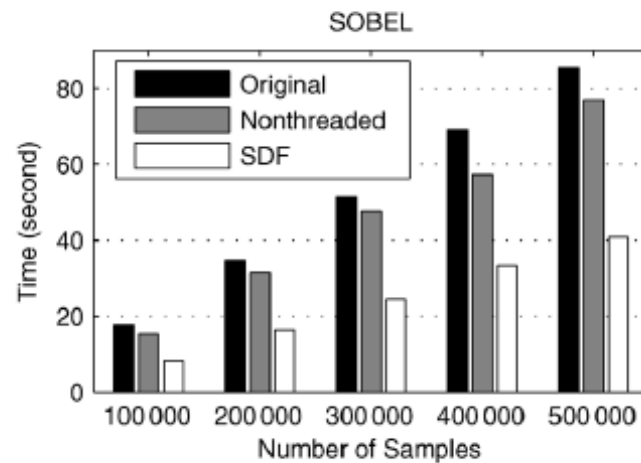
- Purpose: Scientific work ...
- Extends SystemC kernel capabilities
- SDF, FSM, CSP MoC integrated in SystemC-AMS
- MoC can be structured hierarchically
- Lacks support for CT domain
- ... **interesting**: Comparison SDF with + without kernel extension (diophantine solver) →



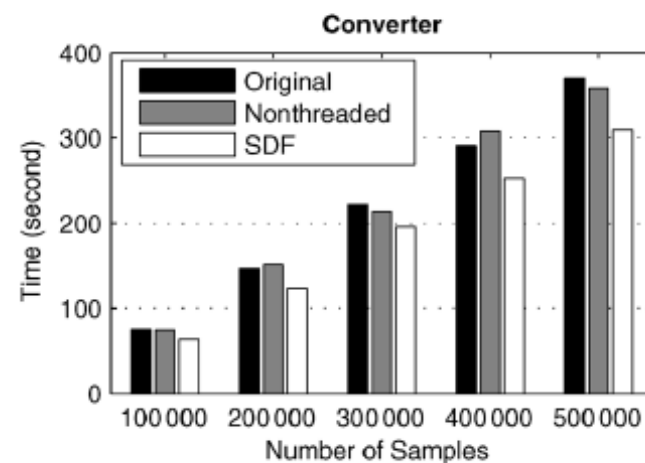
(a)



(b)



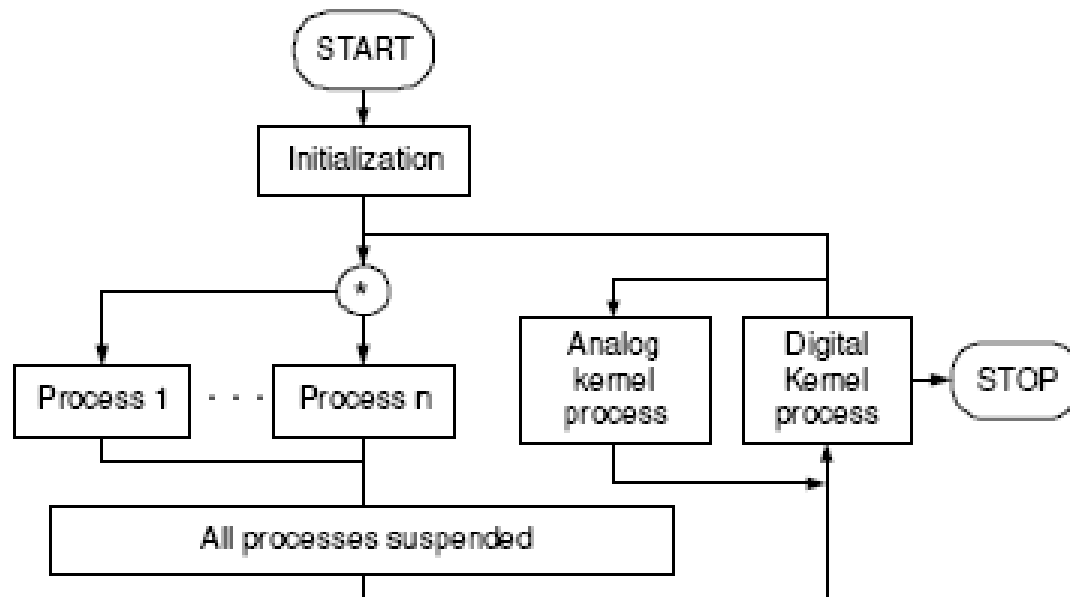
(c)



(d)

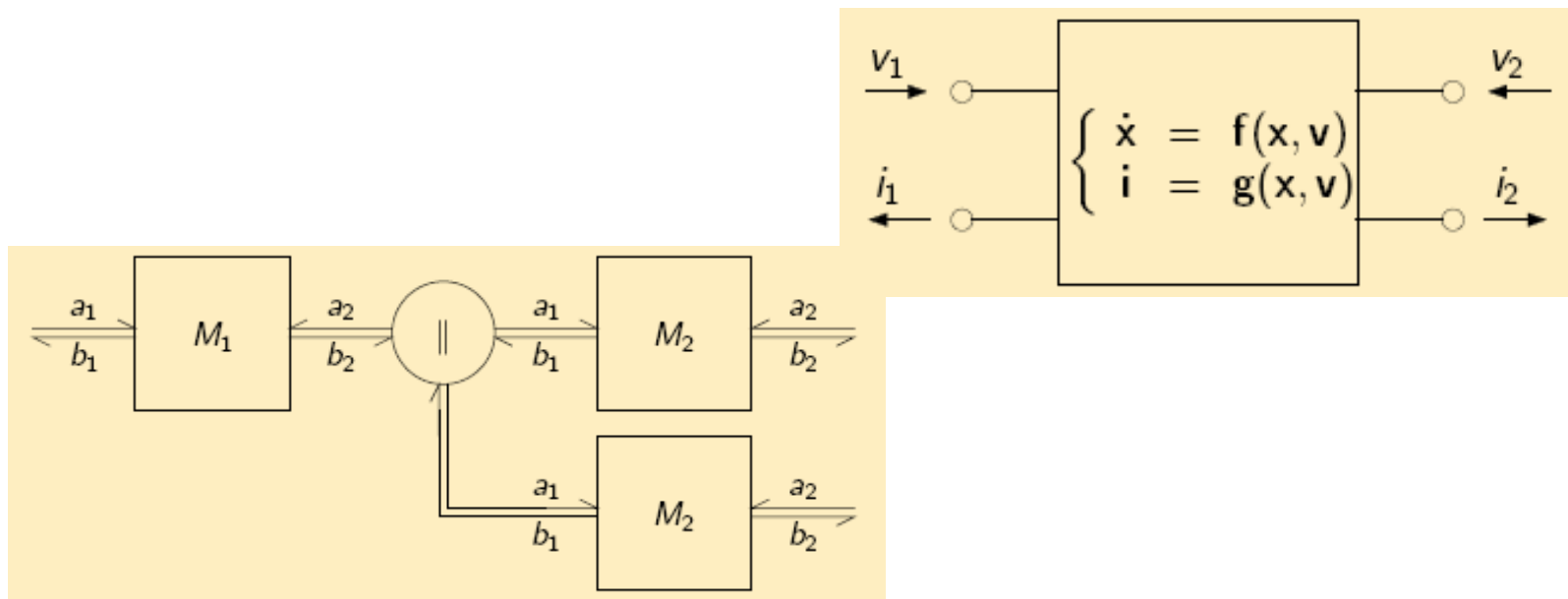
SEAMS [T. Kazmierski et al, 2004]

- 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/>





- 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

- 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)

- 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 ...)

