

ECPS 203

Embedded Systems Modeling and Design

Lecture 11

Rainer Dömer

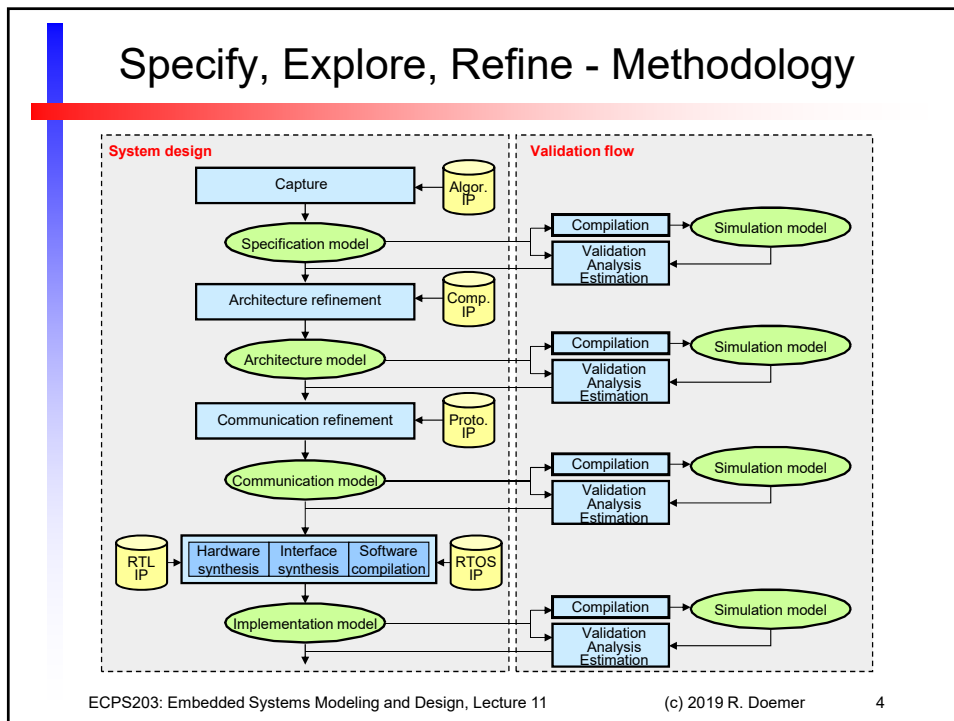
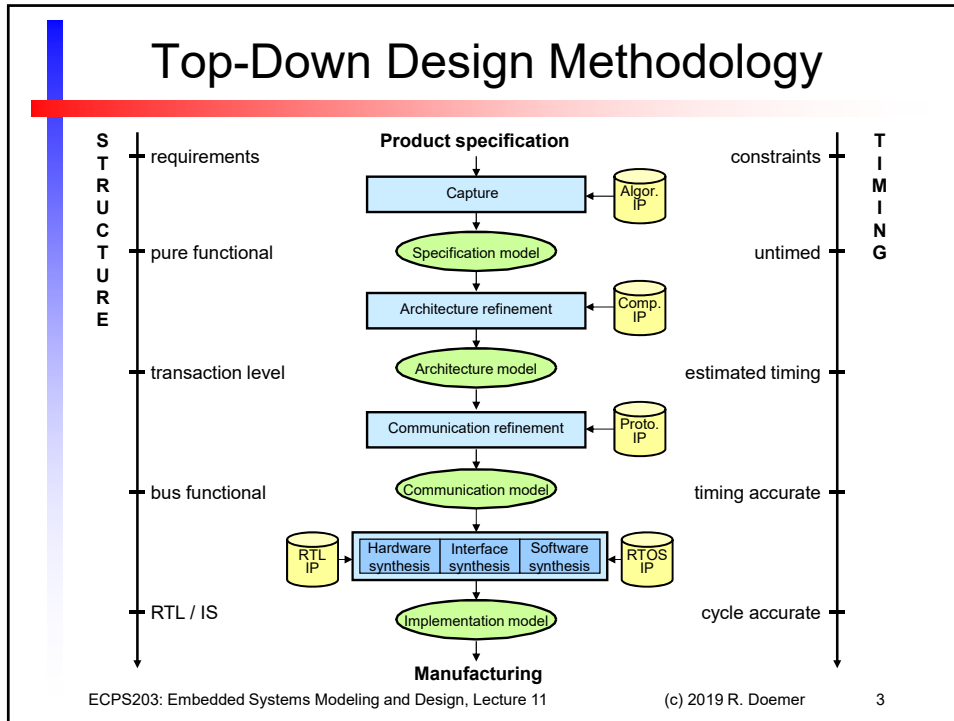
doemer@uci.edu

Center for Embedded and Cyber-physical Systems
University of California, Irvine



Lecture 11: Overview

- Embedded System Design Flow
 - Top-down design methodology
 - Refinement-based design flow
 - Specify
 - Explore
 - Refine
- System-on-Chip Environment (SCE)
 - Application example: GSM Vocoder
 - Interactive demonstration (part 1)



Specify, Explore, Refine - Design Flow

- Refinement steps
 - Architecture refinement (Specification -> Architecture)
 - Communication refinement (Architecture -> Communication)
 - Cycle-accurate refinement (Communication -> RTL/IS)
 - HW / SW / interface synthesis
- Levels of abstraction
 - Specification model: untimed, functional
 - Architecture model: estimated, structural
 - Communication model: timed, bus-functional
 - Implementation model: cycle-accurate, RTL/IS
- Component data bases
 - Algorithms for specification
 - Components for architecture
 - Busses for communication
 - RTOS for SW
 - RTL components for HW

ECPS203: Embedded Systems Modeling and Design, Lecture 11 (c) 2019 R. Doemer 5

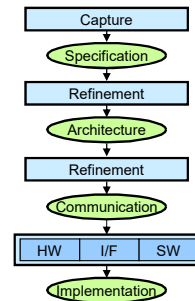
Specify, Explore, Refine - Design Flow

- Refinement Step 1: System Architecture
 - Allocation of Processing Elements (PE)
 - Type and number of processors
 - Type and number of custom hardware blocks
 - Type and number of system memories
 - Mapping to PEs
 - Map each behavior to a PE
 - Map each channel to a PE
 - Map each variable to a PE
- Result
 - System architecture of concurrent PEs with abstract communication via channels

ECPS203: Embedded Systems Modeling and Design, Lecture 11 (c) 2019 R. Doemer 6

Specify, Explore, Refine - Design Flow

- Refinement Step 2: PE Scheduling
 - For each PE, serialize the execution of behaviors to a single thread of control
 - Option (a): Static scheduling
 - For each set of concurrent behaviors, determine fixed order of execution
 - Option (b): Dynamic RTOS scheduling
 - Choose scheduling policy, e.g. round-robin or priority-based
 - For each set of concurrent behaviors, determine scheduling priority
- Result
 - System model with abstract scheduler inserted in each PE



ECPS203: Embedded Systems Modeling and Design, Lecture 11

(c) 2019 R. Doemer

7

System-on-Chip Environment (SCE)

- Integrated Development Environment (IDE) with support of:
 - Graphical frontend (*sce*, *scchart*)
 - SLDL-aware editor (*sced*)
 - Compiler and simulator (*scc*)
 - Profiling and analysis (*scprof*)
 - Architecture refinement (*scar*)
 - RTOS refinement (*scos*)
 - Communication refinement (*sccr*)
 - RTL refinement (*scrtl*)
 - Software refinement (*sc2c*)
 - Scripting interface (*scsh*)
 - Tools and utilities (*sir_list*, *sir_tree*, ...)

ECPS203: Embedded Systems Modeling and Design, Lecture 11

(c) 2019 R. Doemer

8

SCE Main Window

The screenshot shows the SCE Main Window with a project hierarchy on the left and a table of components on the right. The table lists various components and their properties:

| Name | Type | N | Computation [cycles] | De [ct] |
|-----------------|---------------------|------|----------------------|---------|
| Open_Loop | | 163 | 267413 | |
| syn_filter | Syn_Filt | 3912 | 5226 | |
| residual | Residu | 1956 | 5777 | |
| ol_tag_estimate | Ol_Tag_Est | 163 | 222092 | |
| for_init | Open_Loop_Init | 163 | 0 | |
| for_end | Open_Loop_End | 652 | 81 | |
| for_body2 | Open_Loop_Body2 | 652 | 244 | |
| for_body1 | Open_Loop_Body1 | 652 | 1 | |
| wp1 | short int [40] | | | |
| p_speech_l | short int * | | | |
| mem_w | short int [10] | | | |
| i | int | | | |
| A_U | short int [11] | | | |
| sp1 | short int [11] | | | |
| wp | inout short int * | | | |
| txdx_ctrl | in unsigned bits[0] | | | |

ECPS203: Embedded Systems Modeling and Design, Lecture 11

Copyright © 2003 CECS

9

SCE Source Editor

The screenshot shows the SCE Source Editor with a C code snippet for a behavior named 'Coder_12k2_Seq1':

```

behavior Coder_12k2_Seq1
{
  in  Word16 speech_proc[L_FRAME],
      Word16 old_speech[L_TOTAL],
      Word16 *speech;
  out Word16 *p_window,
      Word16 old_wsp[L_FRAME + PIT_MRK()],
      Word16 *wsp;
  out Word16 old_exc[L_FRAME + PIT_MRK + L_INTERPOL],
      Word16 *exc;
  out DTctrl_t txdx_ctrl;
  in  Flag reset_flag;
}

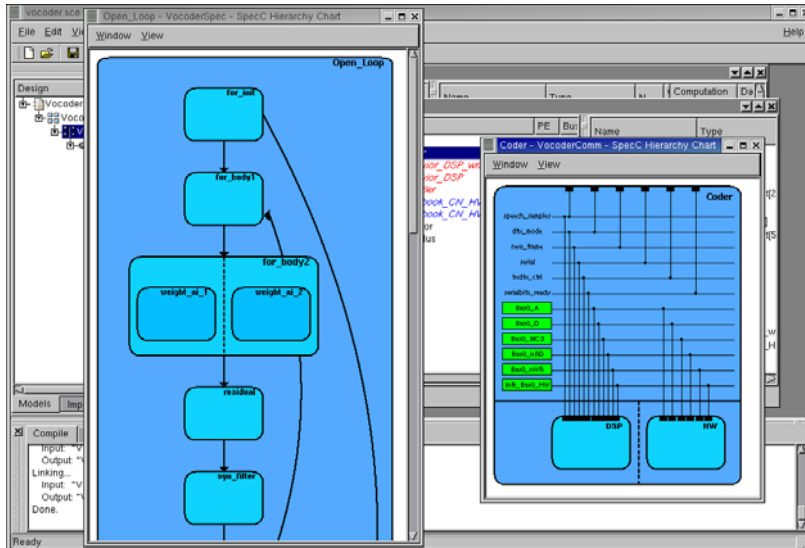
implements Ireset
{
void init(void)
{
  /*----- Initialize pointers to speech vector. -----*/
  speech = old_speech + L_TOTAL - L_FRAME; /* New speech */
  p_window = old_speech + L_TOTAL - L_WINDOW; /* For LPC window */
  /* Initialize pointers */
  wsp = old_wsp + PIT_MRK;
  exc = old_exc + PIT_MRK + L_INTERPOL;
  /* vectors to zero */
  Set_zero (old_speech, L_TOTAL);
  Set_zero (old_exc, PIT_MRK + L_INTERPOL);
  Set_zero (old_wsp, PIT_MRK);
  txdx_ctrl = TX_SP_FLAG | TX_VAR_FLAG;
  ptch = 1;
}
}
    
```

ECPS203: Embedded Systems Modeling and Design, Lecture 11

Copyright © 2003 CECS

10

SCE Hierarchy Displays



ECPS203: Embedded Systems Modeling and Design, Lecture 11 Copyright © 2003 CECS 11

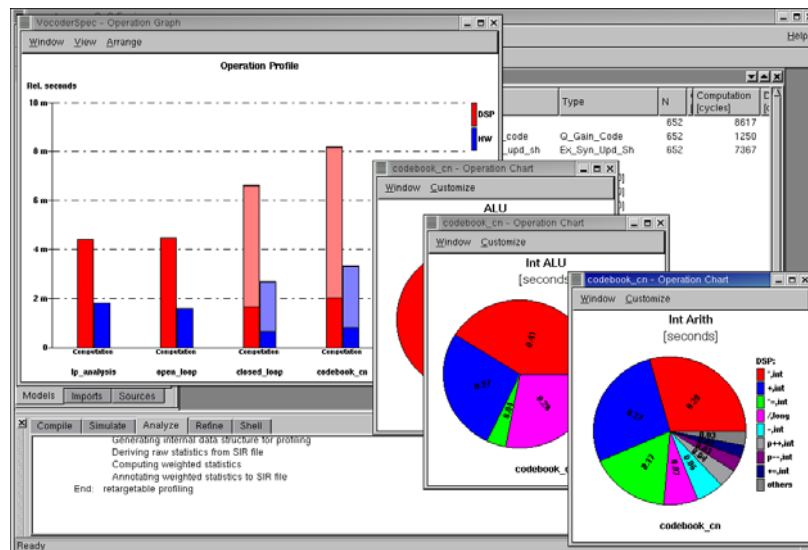
SCE Compiler and Simulator

European digital cellular telecommunications system
 12200 bits/s speech codec for
 enhanced full rate speech traffic channels
 Bit-Exact SpecC Simulation Code - encoder
 Version 1.0
 March 31, 1999

| Frame | encoding delay |
|-----------|--------------------------|
| Frame= 1 | encoding delay = 0,00 ms |
| Frame= 2 | encoding delay = 0,00 ms |
| Frame= 3 | encoding delay = 0,00 ms |
| Frame= 4 | encoding delay = 0,00 ms |
| Frame= 5 | encoding delay = 0,00 ms |
| Frame= 6 | encoding delay = 0,00 ms |
| Frame= 7 | encoding delay = 0,00 ms |
| Frame= 8 | encoding delay = 0,00 ms |
| Frame= 9 | encoding delay = 0,00 ms |
| Frame= 10 | encoding delay = 0,00 ms |
| Frame= 11 | encoding delay = 0,00 ms |
| Frame= 12 | encoding delay = 0,00 ms |
| Frame= 13 | encoding delay = 0,00 ms |
| Frame= 14 | encoding delay = 0,00 ms |
| Frame= 15 | encoding delay = 0,00 ms |
| Frame= 16 | encoding delay = 0,00 ms |
| Frame= 17 | encoding delay = 0,00 ms |
| Frame= 18 | encoding delay = 0,00 ms |
| Frame= 19 | encoding delay = 0,00 ms |
| Frame= 20 | encoding delay = 0,00 ms |
| Frame= 21 | encoding delay = 0,00 ms |
| Frame= 22 | encoding delay = 0,00 ms |
| Frame= 23 | encoding delay = 0,00 ms |
| Frame= 24 | encoding delay = 0,00 ms |
| Frame= 25 | encoding delay = 0,00 ms |
| Frame= 26 | encoding delay = 0,00 ms |
| Frame= 27 | encoding delay = 0,00 ms |

ECPS203: Embedded Systems Modeling and Design, Lecture 11 Copyright © 2003 CECS 12

SCE Profiling and Analysis



ECPS203: Embedded Systems Modeling and Design, Lecture 11

Copyright © 2003 CECS

13

SCE Demonstration

- Application Example: GSM Vocoder
 - Enhanced full-rate voice codec
 - GSM standard for mobile telephony (GSM 06.10)
 - Lossy voice encoding/decoding
 - Incoming speech samples @ 104 kbit/s
 - Encoded bit stream @ 12.2 kbit/s
 - Frames of $4 \times 40 = 160$ samples ($4 \times 5\text{ms} = 20\text{ms}$ of speech)
 - Real-time constraint:
 - max. 20ms per speech frame
(max. total of 3.26s for sample speech file)
 - SpecC specification model
 - 29 hierarchical behaviors (9 par, 10 seq, 10 fsm)
 - 73 leaf behaviors
 - 9139 formatted lines of SpecC code
(~13000 lines of original C code, including comments)

ECPS203: Embedded Systems Modeling and Design, Lecture 11

Copyright © 2003 CECS

14