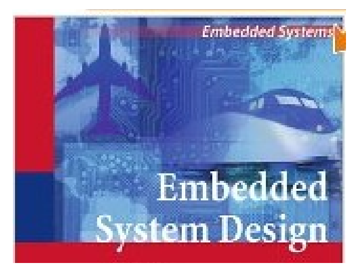
technische universität
dortmund

fakultät für informatik
informatik 12

Embedded System Design 2.0: Rationale Behind a Textbook Revision



Peter Marwedel, Michael Engel
TU Dortmund, Germany

Problem addressed (1)

- Embedded systems frequently not integrated into curricula
- Lecturers of existing courses are faced with problem:
  - Selection of content
    - Difficult for the non-specialist to find a good survey
  - Selection of text book
    - Many of the available text books at a too low level: Programming of microprocessors, memory maps, interrupts, … & rehash of computer architecture topics

Problem addressed (2)

- 1st edition of my own book was published in 2003, not taking recent results and focus shifts into account
  - Little emphasis on models of computation
  - No real-time calculus
  - No computation of WCET
  - No mapping to multi-processors
  - No reference to cyber-physical systems
  - No coverage of reliability

Related work

- P. Caspi, et al.: Guidelines for a graduate curriculum on embedded software and systems, *ACM TECS*, 2005
- Shiao-Li Tsao et al.: The development and deployment of embedded software curricula in Taiwan. *SIGBED Rev.*, 4:64–72, January 2007
- A. Jantsch: *Modeling Embedded Systems and SoC's: Concurrency and Time in Models of Computation,* 2004
- R. Zurawski (ed.): *Embedded Systems Handbook,* 2006.
- D. Gajski et al.: *Embedded System Design*, 2009
- H. Kopetz: *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 2011
- E. A. Lee et al.: Introduction to embedded systems, a cyber-physical systems approach, 2011.

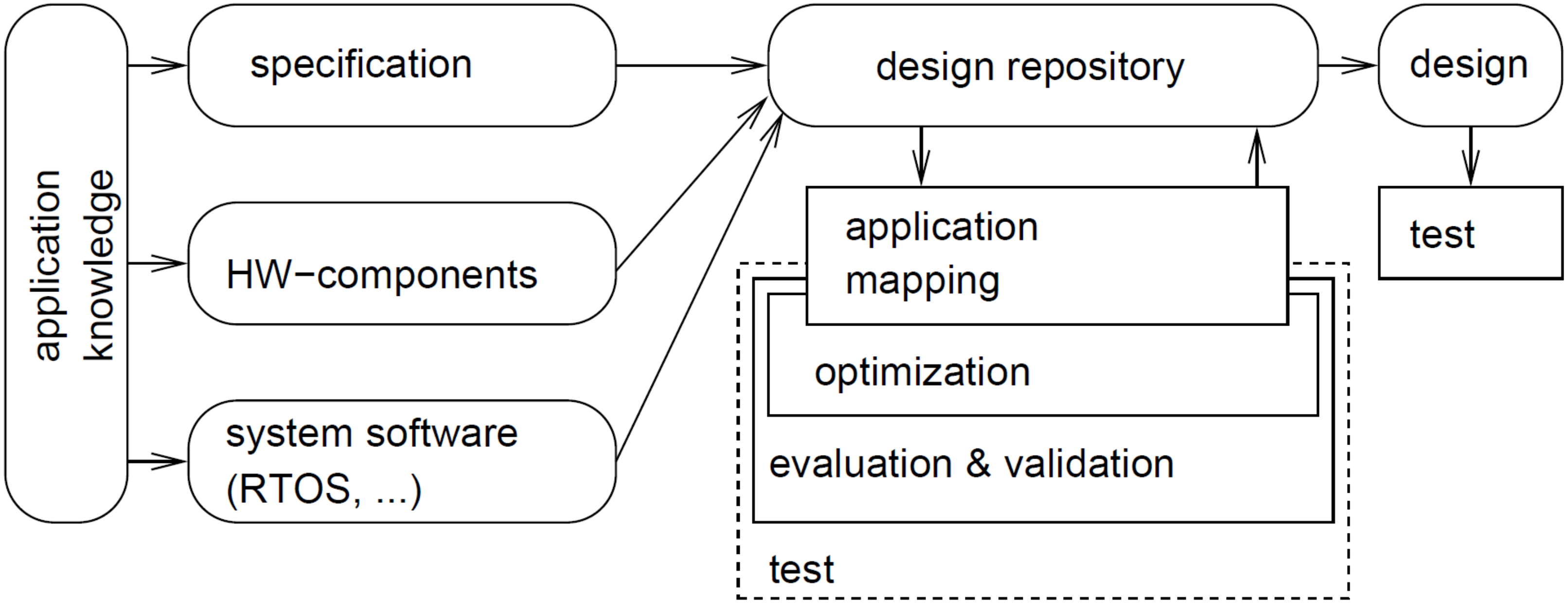
SCOPE

- How to distinguish between embedded systems (ES) and cyber-physical systems (CPS)?
- Position:
  Cyber-physical system (CPS) =
  Information processing (ES) + *physical environment*
- Impossible to cover physical environment at depth
- ES remains relevant by itself, but impact of link to physical environment should be highlighted
- Distinction between small computing platforms (e.g. small phones) and ES integrated into physical environment

Content selection

- Listening to conference presentations
- Reading publications
- Listening to colleagues from industry
- Logical links in the selected material
- Experience with > 10 years of teaching the subject
- Experience with using the 1st edition of the book

Structure



- One chapter each on specification & modeling, HW-components, system software application mapping, evaluation & validation, optimization, test
- Clear structure facilitates integration of custom material

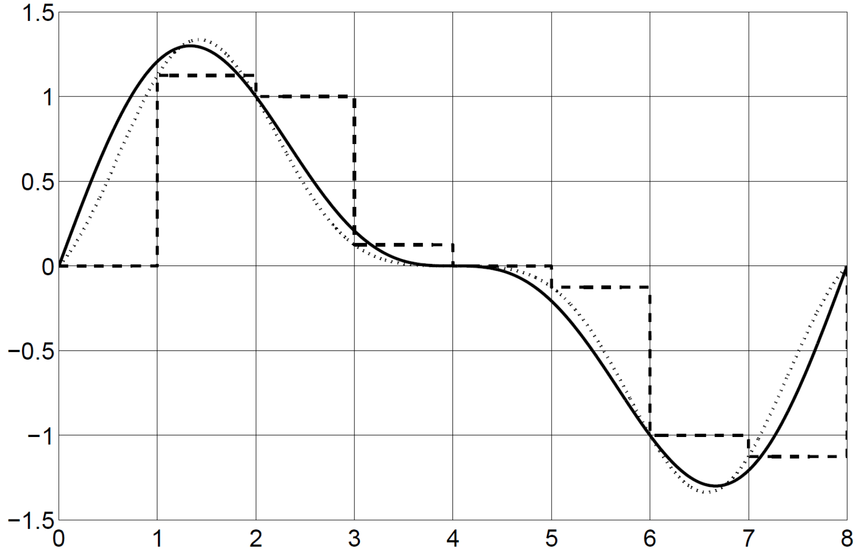
New structure for specification and modeling

Focus shifted from languages to models exemplified by languages

Communic./ Organiz. of components	Shared memory	Message passing: synchronous	Message passing: asynchronous
Undefined components	Plain text or graphics, use cases	(Message) sequence charts	
Communicating finite state machines	StateCharts		SDL
Data flow	(not useful)		Kahn process networks, SDF
Petri nets		C/E nets, P/T nets, ...	
Discrete event (DE) model[1]	VHDL, Verilog SystemC	(Only experimental systems) (Distributed DE in Ptolemy)	
Von-Neumann model	C, C++, Java	C, C++, Java with libraries CSP, ADA	

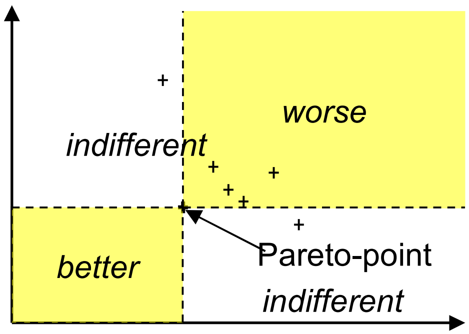
Chapter on HW: Extended coverage of A/D- and D/A-conversion

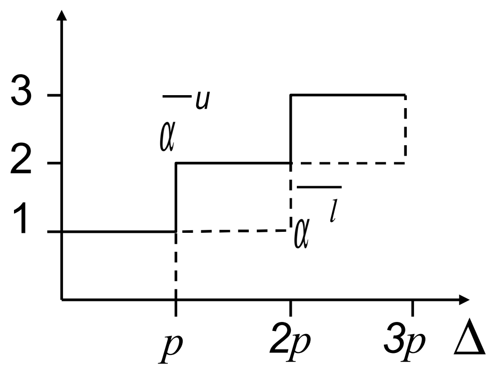
- Observation: difficulties with understanding A/D- and D/A conversion
- Impossible to include full sampling theory
- Signals formally introduced
- Limited reconstruction demonstrated by an example
- Op-amp explained in appendix



Chapter on evaluation and validation

- Evaluation techniques more mature than in 2003
- Focus on multi-objective modeling, added introduction of Pareto-optimality
- Including more objectives than before
  - WCET, real-time calculus
  - Reliability modeling
  - Energy, power
- Close link to validation due to similar techniques being used





Other chapters: system software, optimizations and testing

- Chapter on embedded operating systems changed into chapter on embedded system software
- Optimizations: Clearer than in 2003:

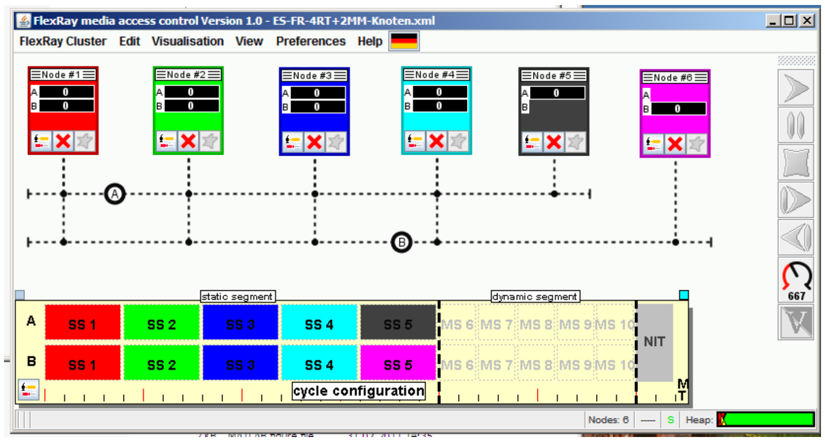
  We can cover only examples of optimizations

  Examples used: Task level concurrency management, high-level optimizations, compilers for embedded systems, power management and thermal management
- Testing:

  Link to testing should be maintained, but this topic may be skipped for shorter editions of the course.

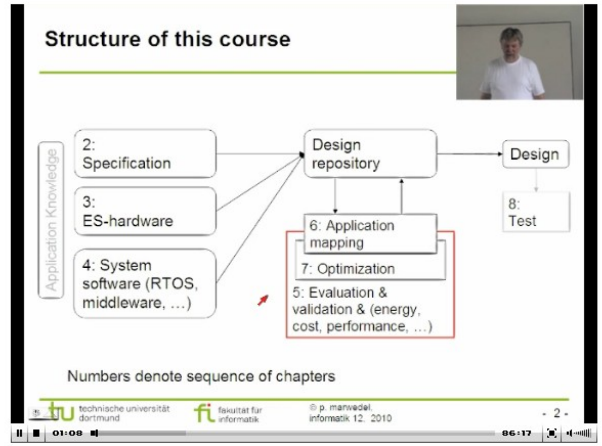
Other extensions

- Integration of simulation software, e.g. for FlexRay® communication



- Video recording

"EMBEDDED SYSTEMS" LECTURE VIDEOS PART 18 (05.07.2010)



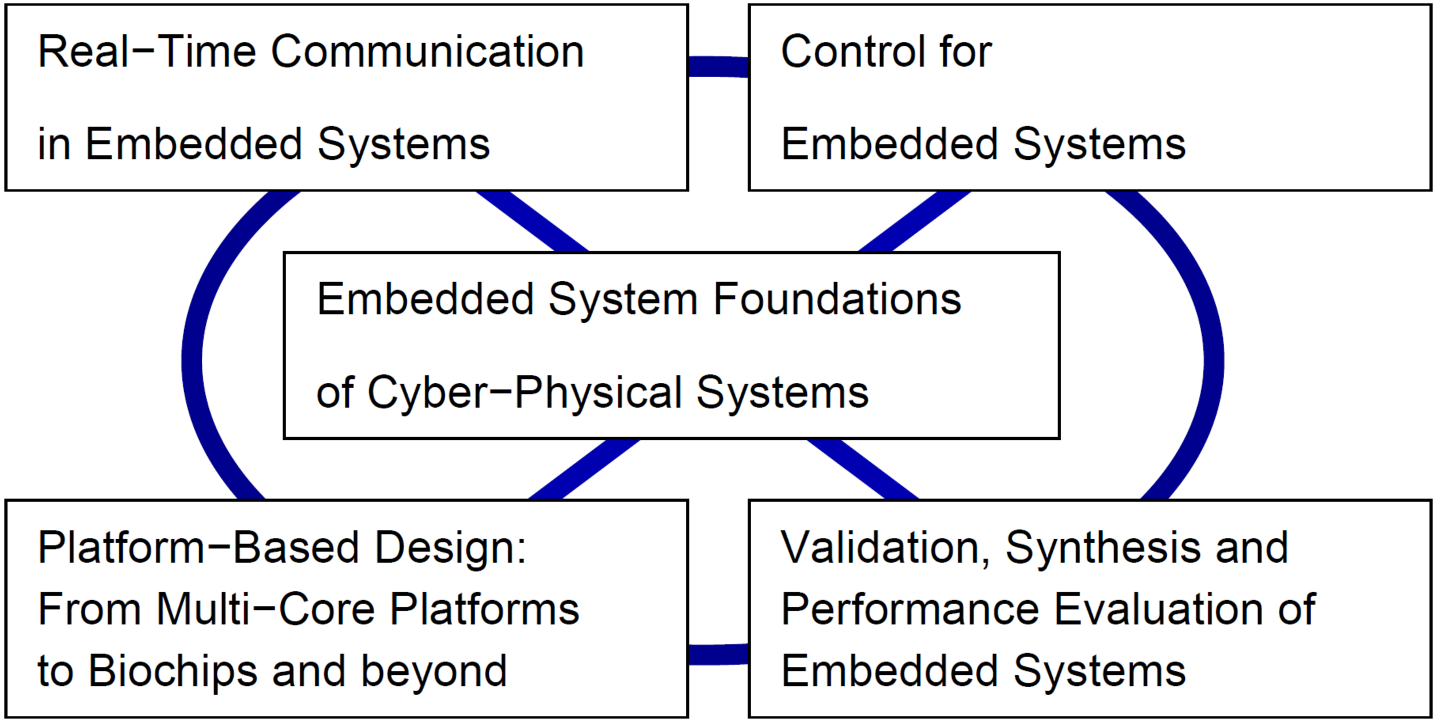
- Assignments

Evaluation (1)

- Translation into local language well received
- Students: Extending coverage of programming ES!
  (conflict between fundamentals and practical training)
- Colleague: More detailed coverage of topics
  (again being limited by available time)
- At Cyprus: Coverage of evolutionary algorithms requested
  (serious description would require copying many pages)
- At several universities: new course required to precede course based on the book
  (book sometimes used at graduate level)
- E. Lee: listing book as complementing his book on CPS

Evaluation (2)

- Defining starting knowledge for anyone working on ES
- Used for 1st day in ES summer school at Beijing in 2011



Summary

- Continuing need to support ES teaching by text book
- Earlier text book needed an update
  - From languages to models exemplified by languages
  - RTOS chapter extended into chapter on system SW
  - More detailed coverage of A/D- and D/A-conversion
  - Introduced chapter on mapping of applications
  - Introduced chapter on evaluation and validation
  - Representative set of optimizations
  - Simulation software, videos, and assigments integrated