Competence Research:

Teaching Embedded Micro/Nano Systems
1. Background
2. Competence Model
3. Current Work
4. Summary
Motivation

– Several years ago, the New York Times estimated that the average American came into contact with 100 microprocessors daily [Wolf et Madsen, 2000]

– "Computer-based embedded systems have been designed for more than 30 years and the need for adequate education in embedded systems is deemed more important now than ever." [Grimheden et Törngren, 2005]

– "Embedded systems are under-represented in teaching and in public discussion" [Marwedel, 2011]
Talks

- Practical Embedded Systems Engineering Syllabus for Graduate Students with Multidisciplinary Backgrounds
- Embedded System Design 2.0: Rationale Behind a Textbook Revision
- Innovative System and Application Curriculum on Multicore Systems
- Teaching Embedded Software Concepts Using Android
- Teaching Cross-Platform Design and Testing Methods for Embedded Systems using DICE
- Bringing Soccer to the Field of Real-Time Embedded Systems Education

Challenges

- Which competencies do the students need
- How to develop these competencies
- How to structure teaching/learning processes
DFG-Project: Competence development with embedded micro- and nanosystems (KOMINA)

Faculty IV: Natural and Engineering Sciences
Department of Electrical Engineering and Computer Science
Didactics of Informatics and E-Learning (Prof. Dr. Sigrid Schubert) [Steffen Jaschke]
Institute of Microsystems Technology (Prof. Dr. Rainer Brück) [André Schäfer]

Faculty of Engineering
Department of Computer Science
Computer Architecture (Prof. Dr. Dietmar Fey)
Project KOMINA

- Computer engineering education
- Theoretically founded courses/tasks
- Embedded systems engineering
  - Focus on micro- and nanosystems (EMNS)

Project of German Research Foundation (DFG): "Competence development with embedded micro- and nanosystems (KOMINA)"
Paradigm shift

- Miniaturization leads to nano-structured components

- New manufacturing processes in a Bottom-Up approach
  - on molecular level (self-assembly, ...)
  - but NOT the field of computer engineers activity

- Consideration of physical constraints on even high levels of abstraction
  - DFG SPP 1500

- Nanotechnology and its effects are not yet part of teaching in Computer Engineering
Outcome orientation / Constructive alignment

(a) An unaligned course.

[Brabrand, C. 2008]
Competence research

- Competencies ≠ knowledge

Competencies

- skills and abilities
- solving problems in variable situations
- motivational, volitional and social willingness
Taxonomy assignment

- which concrete abilities must the students have?
  - Field Programmable Gate Array (FPGA)
  - be able to **understand** a Field Programmable Gate Array (FPGA) based processor.
  - be able to **implement** a Field Programmable Gate Array (FPGA) based processor.
  - be able to **evaluate** a Field Programmable Gate Array (FPGA) based processor.

- define the teachers/systems intention with concrete verbs
  - to state clearly the examination requirements in order to motivate students learning

- Taxonomy of Anderson & Krathwohl (Bloom)
<table>
<thead>
<tr>
<th>Level</th>
<th>Alternative names</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Remember</td>
<td>recognizing, recalling</td>
</tr>
<tr>
<td>2. Understand</td>
<td>interpreting, exemplifying, classifying, summarizing, inferring, comparing, explaining</td>
</tr>
<tr>
<td>3. Apply</td>
<td>executing, implementing</td>
</tr>
<tr>
<td>4. Analyze</td>
<td>differentiating, organizing, attributing</td>
</tr>
<tr>
<td>5. Evaluate</td>
<td>checking, critiquing</td>
</tr>
<tr>
<td>6. Create</td>
<td>generating, planning, producing</td>
</tr>
</tbody>
</table>

[Anderson/Krathwohl, 2009]
Normative proceeding

- Module descriptions
  - Embedded Systems
  - Computer engineering

- Curricula recommendations
  - ACM/IEEE Computer Science Curriculum
  - German Informatics Society

- Collection of competencies

- Cluster
  - Thematic division
  - Solution approaches
Cluster of competence dimensions

- C1: Competencies as preconditions
- C2: Development competencies
- C3: Competencies for multi-level development
- C4: Non-cognitive competencies
<table>
<thead>
<tr>
<th>C1</th>
<th>Competencies as preconditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1.1</td>
<td>Mathematics</td>
</tr>
<tr>
<td>C1.2</td>
<td>Physics</td>
</tr>
<tr>
<td>C1.3</td>
<td>Computer Science</td>
</tr>
<tr>
<td>C1.4</td>
<td>Electrical Engineering</td>
</tr>
<tr>
<td>C1.5</td>
<td>Material Science</td>
</tr>
<tr>
<td>C1.6</td>
<td>English</td>
</tr>
<tr>
<td>C1.7</td>
<td>Scientific work</td>
</tr>
<tr>
<td>C1.8</td>
<td>Learning organization</td>
</tr>
<tr>
<td>C2</td>
<td>Development competencies</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>C2.1</td>
<td>Organization of the development process</td>
</tr>
<tr>
<td>C2.2</td>
<td>Requirement analysis</td>
</tr>
<tr>
<td>C2.3</td>
<td>System design</td>
</tr>
<tr>
<td>C2.4</td>
<td>Implementation</td>
</tr>
<tr>
<td>C2.5</td>
<td>Optimization and Test</td>
</tr>
</tbody>
</table>

**Examples of C2.2 Requirement analysis:**
- are able to explain the relevance of the terms fault tolerance, reliability, and availability. (ACM/IEEE)
- are able to outline the range of methods for implementing fault tolerance in an operating system. (ACM/IEEE)
C 2.3: System design:

- remember formal- and computer-aided design methods for the design of embedded systems
- understand most important technologies and concepts for designing and analyzing computer-aided systems
- understand structure and function of all important basic circuits and arithmetic logic units
- understand computer systems as stratified abstract machines
- understand, analyze unknown circuits, create own circuits
## C3 Competencies for multi-level development

<table>
<thead>
<tr>
<th>C3.1</th>
<th>Top-Down-Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3.2</td>
<td>Bottom-Up-Design</td>
</tr>
<tr>
<td>C3.3</td>
<td>Meet-in-the-Middle-Design</td>
</tr>
<tr>
<td>C3.4</td>
<td>Yoyo-Design</td>
</tr>
</tbody>
</table>

### Diagram:
- **Abstraction**
  - Top-Down
  - Bottom-Up
- **Specialization**
  - Meet-in-the-Middle
  - Yoyo
C4  Non-cognitive competencies

C4.1 Attitudes
C4.2 Social-communicative competencies
C4.3 Motivational and volitional skills

focused competencies

C1 competencies as precondition

C4 non-cognitive competencies
First results

- Empirical refinement of the competence structure model

- The experts rate the importance of the given competencies of the NCSM:
  - Very important
  - Rather important
  - Rather unimportant
  - Very unimportant

- Survey of 96 experts
- 36 results
First results

C 4.3 Motivational and volitional skills  (1.1-1.4)
C 2.1 Organization of the development process   (1.4)
C 3.2 Bottom-Up-Design         (1.4)

...  

...  

...  

...  

C 3.2 Bottom-Up-Design         (2.7)
C 1.5 Material Science         (3.1)
Alignment of the hardware-practical course

- Current status
  - 16-Bit microprocessor on FPGA

Objectives

- Provide a wider range of competencies (C1-3)
  - Focus on the most important competencies
- Project-group based work (C4)
- Life-world related tasks like Android programming (C4)
FPGA-Online

- Reduce cost
- Independent of time and place
- Competence development
Competence structure model

- Normative developed
  - C1-C4
- Empirical refinement in progress (Dec. 2011)
  - The most defined sub competencies are important

Course design

- Analyze of existing courses
- Alignment of concepts to the competence structure model

Further work
Thank you for your attention!


