Naturwissenschaftlich Technische Fakultät

Didaktik der Informatik und E-Learning

UNIVERSITÄT SIEGEN

Enhancing the Comparability between Didactic Research on Embedded Systems

Steffen Büchner,

Steffen Jaschke,

Sigrid Schubert

University of Siegen, Faculty IV
Chair for Didactics of Informatics
and E-Learning

Hölderlinstrasse 3,
57076 Siegen-Weidenau
http://www.uni-siegen.de

Table of Contents

Introduction & Motivation

- Competences on Embedded Systems development are required in a variety of areas (automotive, aerospace, consumer electronic, etc.)
- Embedded Systems education is done for freshman, advanced students and graduates alike
- Embedded systems' technical realization differs greatly (Microprocessors, FPGAs, ASICS, etc.)
- Learning units include, e.g.: design-, implementation-, requirement-gathering- or optimization-tasks (and more)
- Not every of those aspects is stated in educational papers

Solution

State clear what type of embedded or cyber-physical system you are talking about and your intention for using it.

- We created a taxonomy including at least four viewpoints and aspects of major importance:

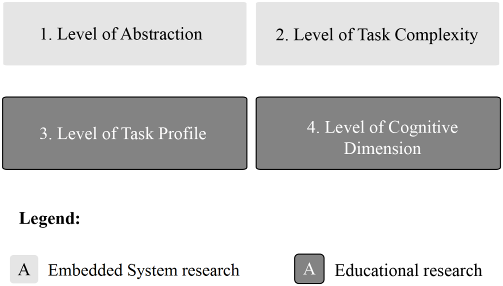


Figure:

Nr. 1.: Level of Abstraction

- Used to describe the level of access to system functionality
- For example: Libraries, IP-cores or AND/OR-Gates
- Does *not* relate to programming language or implementation technique ...
- ... but the level on which the system is developed

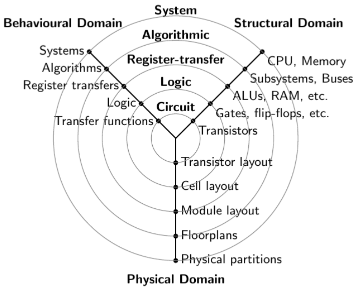


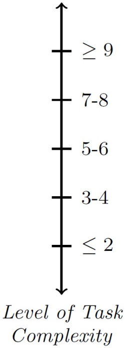
Figure: Gajski Kuhn chart [1]

Nr. 2.:Level of Task Complexity

- Complexity is a difficult term with multiple definitions available
- Best described with common characteristics (our opinion)
- An embedded system is complex if it:
    - controls (rather then only gets) environmental attributes
    - controls physical, biological or chemical parameters
    - uses parallelism
    - needs to meet standards
    - has a very large state space/huge amount of parts
    - uses real time operations
    - requires strong ergonomic or non-functional requirements (e.g. size, safety)
    - to be continued ...

Nr. 2.:Level of Task Complexity

- Every of those characteristics is one point
- By adding all of those points, one can describe how complex the system under observation is
- Not nearly accurate enough for any *real-life* project ...
- ... but reasonable accurate for *educational projects*
- Can be extended and modified without breaking the concept

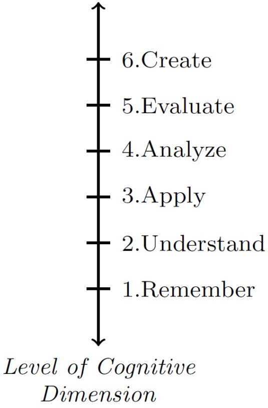


Nr. 3.: Level of Task Profile

- Development workload depends not only on complexity but on task profiles, too.
  1. Build a system virtually from scratch (no existing subsystems, no previous work). No support like code-skeletons or libraries.
  2. Build a system by the use of already existent and completely new parts/functionalities.
  3. Mixture of building new parts by interconnecting existing subsystems and parameterizing them to fulfill a specific task (customizing).
  4. Interconnection of already existing subsystems without any type of adaption to the parts/functionalities themselves (puzzle-like).

Nr. 4.: Level of Cognitive Dimension

- In a educational setting, competences are very important
- Synonymous to the question: *“What is the tasks’ intention?*
- Mapping to the most common taxonomy (Anderson/Krathwohl)[2]



Nr. 4.: Level of Cognitive Dimension

- Taxonomy of A & K is not computer science specific, but nicely applicable
- Taxonomy of Fuller uses the same six levels in a 2-Dimensional order and computer science specific
- Difference e.g.: *creation* of sourcecode $\neq$ level *create*
- Proposal: Try using specific taxonomy on a one dimensional scale

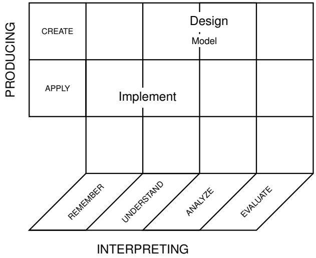


Figure: Taxonomy of Fuller[3]

Example of the Taxonomy (Mitsui et al., 2009)

The project described aims to build "*[...] the basic functions of JPEG encoder, which is widely used for compressing digital images [...]*" [4, p. 4]. This encoder had to be designed as a hardware module for an FPGA. The design language used was no pure *hardware description language* (HDL), but SystemC.

Level of Abstraction

- Use of SystemC often means high abstraction level
- Design has been synthesized to get a net-list for FPGA
- Therefore: Two different marks

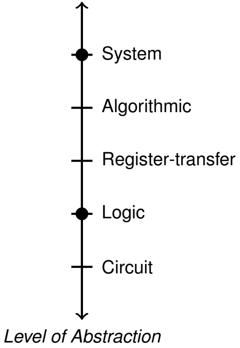


Figure: Level of Abstraction for Mitsui et al.,2009

Level of Task Complexity

- Task requires the development of a JPEG-compliant module
- While parallism is considered, no other "complex"-characteristics are present
- Additionally, the transformation algorithms are given
- Overall, complexity is rather low (ca. ~ 1)

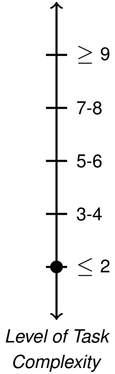


Figure: Level of Task Complexity for Mitsui et al.,2009

Level of Task Profile

- Students do not need to build system from scratch
- Relies heavily on predefined components
- Those get connected and parameterized
- Therefore characteristics of a "puzzle-like" approach

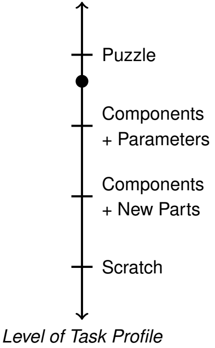


Figure: Level of Task Profile for Mitsui et al.,2009

Level of Cognitive Dimension

- Focus on Design and Implementation
- According to Fuller [3]:
  - Design is a creative task thus, level 6 of A & K taxonomy
  - Implementation is on level 3

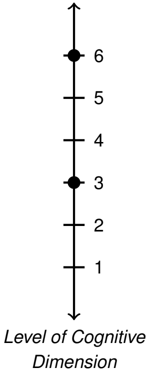


Figure: Level of Cognitive Dimension for Mitsui et al.,2009

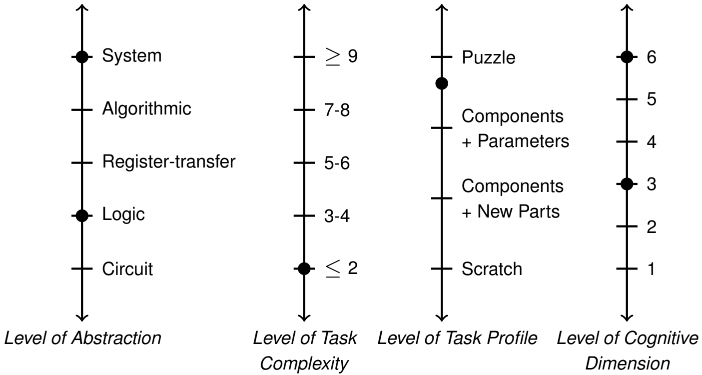


Figure: Taxonomy assignment of Mitsui et al. ([4])

Summary

- Taxonomy with four parameters which supports the comparability of embedded system courses and labs
- Well known classification approaches like the Gajski-Kuhn chart and the taxonomy of Anderson & Krathwohl have been adopted to the taxonomy proposed
- We analyzed three papers und Hardware/Software Co-Design
- Ease the discussion, provide a better reuse, and economize work

Thank you for your attention

D. Gajski and R. Kuhn, “Guest editors’ introduction: New vlsi tools,” *Computer*, vol. 16, no. 12, pp. 11–14, 1983.

L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock, *A taxonomy for learning, teaching, and assessing: A revision of Bloom’s taxonomy of educational objectives*. New York: Longman, abridged ed., [nachdr.] ed., 2009.

U. Fuller, C. Riedesel, E. Thompson, C. G. Johnson, T. Ahoniemi, D. Cukierman, I. Hernán-Losada, J. Jackova, E. Lahtinen, T. L. Lewis, and D. M. Thompson, “Developing a computer science-specific learning taxonomy,” *ACM SIGCSE Bulletin*, vol. 39, no. 4, p. 152, 2007.

H. Mitsui, H. Kambe, and H. Koizumi, “Use of student experiments for teaching embedded software development including hw/sw co-design,” *Education, IEEE Transactions on*, vol. 52, no. 3, pp. 436–443, 2009.