X World Conference on Computers in Education
July 2-5, 2013; Toruń, Poland

Towards a Didactic System for Embedded System Design in Higher Education

Steffen Jaschke, *steffen.jaschke@uni-siegen.de*
University of Siegen, Didactics of Informatics and E-Learning, Siegen, Germany

Abstract
Despite an obvious importance, the research on the educational aspects of embedded systems in higher education is still insufficient. Among others, it lacks a fine grained structure of competences and their interconnections as a starting point. Resulting from a content analysis, competences of system design are differentiated, and described using the category *models* as an example. This refinement serves as a basis for the *didactic system for embedded systems design in higher education*. Within this research step the didactic systems for *object oriented modeling* and *internetworking* both in secondary education were analyzed. Beside the transfer to the specific area of embedded system design, there is a general need for adaptations. The exploration of cognitive structures, the classification of tasks, and the development of associated learning aids will close further research gaps and form the new didactic system.

Keywords
embedded systems, didactic system, modeling, competence research

INTRODUCTION

Embedded Systems (ES) are omnipresent in our everyday life and, at the same time, an important economic sector. They are characterized as information processing systems linked to the physical environment via sensors or actuators including communication. Embedded in products like cars, planes, and telecommunication equipment they have special requirements and constraints like real-time conditions, minimal power consumption, and particularly reliability. This importance is reflected in the need for highly qualified developers of such systems. The qualification regularly takes place in electrical engineering-, computer science-, computer engineering-, or embedded systems specific- study programs. While these different origins result in a variety of specialists and generalists, at the same time, the targeted levels of competences differ as well as the students' previous knowledge.

Despite the obvious importance the research on the educational aspects of embedded systems in higher education is still insufficient. The reasons for that are manifold. For example, curricula recommendations, textbooks, and module descriptions do not consider competence- and outcome orientation in a sufficient scope (yet). Another point is that insights in, mostly, small groups of test persons of different institutions with regional particularities and a variety of study programs do not allow general statements. Nonetheless, the author states clear, that trends can be made visible even if research results are not generalizable infinitely.

The author's research is part of the project *Competence Development with Embedded Micro- and Nanosystems (KOMINA)*, promoted by the *German Research Foundation (DFG)*. In this project basics of computer engineering's didactics are explored. This includes future technologies like nanostructured devices (Kleinert et al., 2012).Within the project's first research steps a competence structure model has been developed through a normative proceeding and was empirically refined subsequently through a survey in which 171 experts were asked to rate the importance of the normative derived competences. As a result the structure of four competence dimensions is given in Table 1 (Schäfer et al., 2012).

Table 1: Empirically refined Competence Structure Model (ECSM)

C1	**Preconditions**	**C2**	**Development Competences**
C1.1	Mathematics	C2.1	Organization
C1.2	Physics	C2.2	Requirements analysis
C1.3	Computer science	C2.3	System design
C1.4	Electrical engineering	C2.4	Implementation
C1.5	English	C2.5	Optimization and test
C1.6	Scientific work		
C1.7	Learning process		
C3	**Multi-level Development**	**C4**	**Non-cognitive Competences**
C3.1	Top-down design	C4.1	Attitudes
C3.2	Bottom-up design	C4.2	Social-Communication S.
C3.2.1	Nano effects	C4.3	Motivational & Volitional S.
C3.3	Jojo		

Now, in the author's research part, the focus is on:

1. the differentiation of embedded system design competences
2. the refinement of competences towards cognitive structures
3. the classification of tasks
4. the development of learning aids

Referring to Brinda and Schubert (2001); Brinda (2004); and Freischlad (2008, 2010) one can subsume the last three points into the development of a *didactic system* whereas point one is a precondition for competence orientation. Didactic systems can be described as an open compound of *knowledge networks*, *exercise classes*, and *exploration modules* that provide different beneficial didactical functions (Brinda & Schubert, 2001). Within this paper, the groundwork (point one) towards the didactic system is presented. This includes the analysis of existing didactic systems as well as the reasoning for adaptations.

Related work and the research conditions are described in the following section. Preceding research about didactic systems and its implications are discussed afterwards. The methodology towards a didactic system for embedded system design is described. After concluding, the author's next research steps are shown in the further work section.

RELATED WORK

Since the topic of embedded systems is much too broad to give general didactical recommendations for all branches, rather generalizable concepts are to be explored. This includes, in particular, the first teaching at the universities' bachelor programs. At a later time, the programs offer too many possibilities for specialization. The target groups (in the sense of probands) of this research part are undergraduate students with computer science background. They actually have not enrolled specifically for an embedded systems program. Instead, through the combination of computer science and electrical engineering the required motivation and attitudes can be assumed.

The empirical research takes place at one particular laboratory course called *Design and Application of Embedded Microsystems*. Here, the project group (Schäfer et al., 2012) designed an introductory embedded systems course with a focus on hardware basics and the implementation of a home automation system that consists of actuators (LED, peltier element, fans) and sensors (brightness, humidity, temperature indoor, temperature outdoor, shock sensor). The students implement the control unit for both microcontrollers and Field Programmable gate Arrays. A first run of this course accompanied by an extensive observation was performed in summer 2012 (Jaschke et al., 2012). In order to evaluate this course, among others, the author's research results get incorporated. Until now, the course's tasks are linked to competences of the ECSM (Table 1) like this:

> ”Build the brightness-, humidity-, and temperature sensor (C1.4) of your prepared circuit (C2.2, C2.3) from preparation task 14/15 on the breadboard. Check roughly whether all sensors operate as expected (C2.5) (Jaschke et al., 2012).”

Neither the required preconditions nor the targeted levels of competences are clearly recognizable. This is due to the fact that the ECSM provides a structure and a weighting in general, but no detailed insights. Thus, the students lack an orientation in the educational process. They do not recognize which tasks are to be performed to enhance one of the rough competences from the ECSM. In order to provide orientation by means of a didactic system, first, a limitation on one competence dimension is needful.

In general, referring to ACM/IEEE (2005) the students' superior competences that should be reached can be formulated as follows:

> ”computer engineers should be able to *design* and *implement* systems that involve the integration of software and hardware devices [emphasis added]”

These two broad competences are to be explored as they are of major importance for tomorrow's embedded system developers. This is also confirmed by available textbooks and recommendations that primarily deal with *system design (C2.3)* or *implementation (C2.4)*. The latter usually can be found in literature which relates to the mere programming of embedded systems only, but not address their holistic development. Thus, the pre-, post-, and accompanying activities of the development process are not considered in this research. Moreover, here, the focus is on embedded systems' design, only.

Competences

The *Empirically refined Competence Structure Model (ECSM)*, developed within the KOMINA project differentiates four dimensions of competences roughly. It has been developed on the basis of curricula recommendations, universities' module descriptions and textbooks regarding embedded systems or computer engineering. According to Weinert (2001, p. 27 f), competences are defined as:

> ”the cognitive abilities and skills that individuals possess or can learn for solving specific problems, and the associated motivational, volitional and social readiness and abilities that enable them to use these solutions responsibly and successfully in a variety of situations.”

As one can see, the term competence is a multidimensional complex structure of skills, abilities and non-cognitive components. As already mentioned, the rough structure of competence dimensions of the ECSM is not sufficiently suitable to be used for the planning of activities, courses, curricula, or corresponding assessments in detail. Instead, it gives a rough insight into the distribution of competences. The ECSM do not provide any links that are necessary for the description of learning paths that include different levels of competences. This is due to the fact that it does not represent any dependences and interconnections of competences, except the dimension C3 which includes those in particular. This result in a need for research as it is an important basis of the didactic system.

Taxonomy
In order to discuss competences of embedded system design the author considers the taxonomy of Anderson et al. (2000) and its computer science specific modification (Fuller et al., 2007). Here, the typical activities producing and interpreting are represented as own dimensions. Although this was originally done for computer programming, this separation is also meaningful for embedded system development. While design can be located in the category Create/Analyze, implement requires a lower cognitive level (see Figure 1).

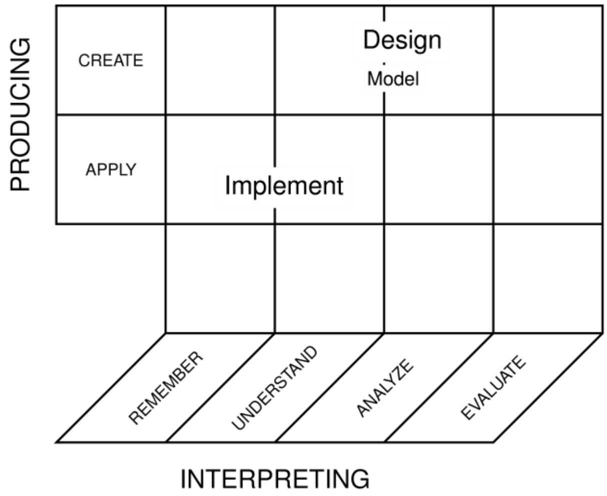

Figure 1: Computer Science specific taxonomy

For implementing a given solution as a design, developers need to apply the processes of implementation and to remember/understand algorithms, languages and so on (Fuller et al., 2007). Design, however, requires more analytical and planning skills. Both activities design and implementation are largely independent and do not imply a sequence within educational processes.

DIDACTIC SYSTEMS

Brinda and Schubert (2001) introduced the concept didactic system as an open compound of *knowledge networks*, *exercise classes*, and *exploration modules*. While Brinda (2004) focused on *object oriented modeling*, Freischlad (2010) discussed and transferred the didactic system to the area of *internetworking*, both in secondary education. The didactic system for embedded system design in higher education is visualized in Figure 2.

The terminology of the systems' components differs from previous research. This results from a need for adaptations described in this section (from → to). Competence research as well as already developed hardware and software within the KOMINA project serves as a basis for the system and, together, will support the competence development of future experts.

Knowledge Networks → Cognitive Structures

Knowledge networks represent didactic metaknowledge that supports the orientation of students within the process of learning, the organization and planning of teaching/learning, and the discussion about these processes while making didactic decisions visible (Freischlad, 2010). Subject specific concepts have to be explored in order to structure knowledge and to reveal knowledge-units. Within the development of these structures it is important not simply to copy the subjects' structure and concepts to educational processes. Freischlad (2010) justifies this with layers of the ISO/OSI layer model where some services have no life world relation to students. For embedded systems this becomes clear when considering a variation of the Gajski-Kuhn chart (Gajksi, 1983) commonly used to describe hardware design processes. This diagram visualizes three domains (behavior, structure, and geometry) on five abstraction levels (from circuit to system). A sequential educational process following the levels of each domain and discussing its concepts and technologies seems not to be useful. Especially the levels'- and the domains' dependencies are of major importance in the development process. Design decisions on one level effect parameters on other levels in the model. As a consequence of this a constant alternation of levels is needed. This was described as competences for multilevel-development (C3, see Table 1) (Jaschke et al., 2011). Following from both, subject specific perspective and didactic perspective, sequential processes do not represent the subject's properties adequately. The levels are neither of same importance within all application areas/design processes nor built up from a common ground of necessary and previously achieved competences.

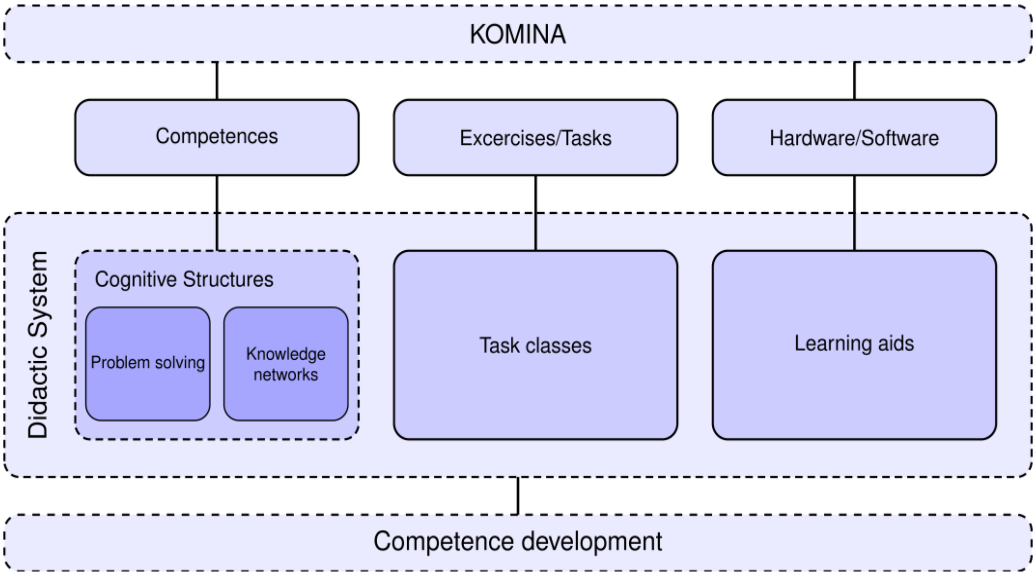

Figure 2: Didaktic System

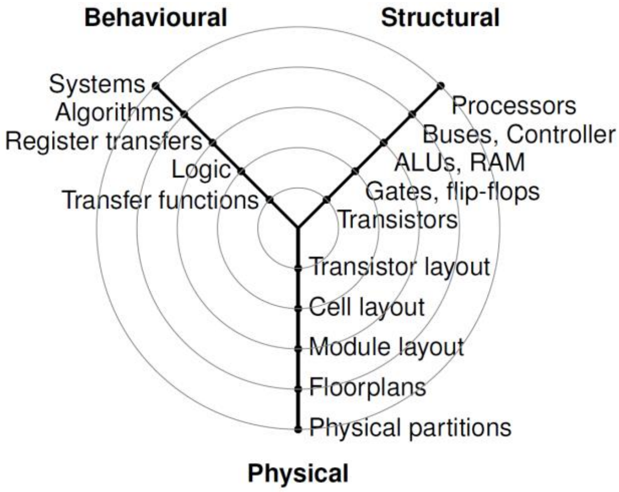

Figure 3: Gajski-Kuhn chart

All parts of the didactic systems (Brinda, 2004; Freischlad, 2010) have in common that they are described in detail as they are designed for specific teaching units. For the transfer to the topic embedded system design a similar granularity has to be achieved. Moreover, a stronger consideration of research results about competence orientation in computer science is necessary. Therefore, a limitation on knowledge structures is insufficient. A holistic consideration of competence development and cognitive structures has to be targeted. This approach also implicates a stronger connection to *solving specific problems* of the term competence. Precisely because problem solving is a key aspect of both, the term competence and the engineering discipline, cognitive structures have to be explored first.

Exercise Classes → Task Classes

Exercise classes describe a classification that supports the construction and selection of exercises as well as the orientation in the set of available and missing ones. In the author's research part, there is a stronger relation to competence orientation and problem solving. Thus the term task is more reasonable as it implies a relation to a context. These tasks can be distinguished according to their purposes in competence development and competence measurement. The latter one requires the development of empirical verified test-items (Schubert & Stechert, 2010).

Exploration Modules / Learning Software → Learning Aids

Exploration modules are learning aids that, in most cases, are embedded in software that supports action-oriented learning. While Brinda (2004) did research about exploration modules that support the learning of some concepts independently, Freischlad (2010) focuses on a collection of scenarios within one particular environment. This environment supports the exploration of internetworking applications and services. In contrast to internetworking, embedded systems cover hardware related topics where exploratory learning is well known e.g. within laboratory experiments. Among others, because of safety restrictions there is still a limitation of free exploration. Providing simulations for getting first experiences with hardware and equipment is essential if a high degree of freedom is intended. That's why in the author's research the third part of the didactic system is more general (learning aids) and includes exploration modules within hardware and software environments. In any case, through the development of exploration modules the application of trial-and-error strategies should be avoided. It is rather preferable is to hypothesize and to apply problem-solving strategies.

Content Analysis

In order to close the previously described research gap (differentiation of competences) as a prerequisite for the exploration of cognitive structures, now, literature is analyzed with focus on design (C2.3). Explicitly occurring didactic knowledge is incorporated in the didactic system and will be discussed in terms of its contribution to the structure of learning processes.

Textbooks

The textbooks *Embedded Systems Design (Marwedel, 2011)* and *Introduction to Embedded Systems (Lee & Seshia, 2012)* do provide an introduction to embedded systems and its design process for undergraduate students. A particularity of these publications is the authors' explicit positioning to the educational processes and their reasoning for the structure of the books (Marwedel & Engel, 2011; Marwedel, 2005). Furthermore, they focus on the mainly non-volatile basics as models, principles, and abstractions instead of (today's) common technologies. Due to the targeted competence orientation this proceeding is meaningful, but has to be expanded with technology aspects in order to enable the students to work in industry as well as on laboratory tasks. The analysis is limited to the identification of competence descriptions within the books' and chapters' prefaces.

Cirricula guidelines and -recommendations

Curricula recommendations covering embedded systems in higher education are, in most cases, not formulated with respect to competence/outcome orientation. On the one hand, joint task forces of the *Association for Computing Machinery (ACM)* and the *Institute of Electrical and Electronics Engineers (IEEE)* provide a fine grained body of knowledge and mention objectives only in passing (ACM/IEEE, 2008). The same is valid for the *German Informatics (GI) Society's curriculum* (GI, 2011) and the *Advanced Real-Time Systems (Artist) education group* (Artist Education Group, 2003; Caspi et al., 2005). On the other hand, some task forces focus on mandatory learning outcomes as well as their association with knowledge units (ACM/IEEE, 2004, 2005). Here, the term outcome is not used in its actual meaning and still is from a teaching perspective:

> "A basic outcome of the experimental part of the curriculum for real-time computing should be to show students that real-time systems introduce additional difficulties in system design and programming (Caspi et al., 2005)."

The authors didn't mention what the students are able to do afterwards the teaching unit. The teacher shows and introduces a topic in an undefined manner. Nevertheless, in some cases the author's intention is obvious or already formulated as a learning outcome.

C2.3 System Design

Since competences of embedded systems design require competences of interdisciplinary domains, it is not meaningful to break them down from embedded systems' design to principles of each domain. Instead, embedded systems' specific competences within such a model have to deal with the topic's particularities. Design decisions regarding hardware and software (co-design) are located at (C2.3), however, understanding Kirchhoff's laws don't. Such can be seen as prerequisite competences (C1.4, see Section 1). The cognitive level and composition of competences as preconditions has to be explored and discussed in order to get a finer delimiting.

As a result of the analysis, competences of system design are differentiated, and described using the category system modeling as an example (see Table 2).

Table 2: C2.3 System Design

		Students should be able to
C2.3.1	Models	apply techniques of modeling like schematic diagrams and description languages to create models for a range of problem areas
C2.3.2	Architectures	contrast architectures and recognize their strength, weakness and relationships
C2.3.3	Technologies	understand the fundamentals of embedded software, -hardware, -communication, and interfaces to the physical environment
C2.3.4	Constraints	evaluate constraints and to analyze trade-offs
C2.3.5	Concurrency/ Timing	understand concurrency and timing issues
C2.3.6	Tools	use hardware and software tools

The profile is a summary of anchored examples (AE) that exemplifies the proceeding of the determination of competences (see Table 3).

Models

In some cases the terms system modeling and system design are used synonymously. In order to be more precise, the author uses design when discussing the holistic competence (C2.3) which means the division of solutions, whereas modeling is used when discussing competences with respect to models as abstractions. Due to embedded systems' complexity modeling is essential. While implementing simple control mechanisms on microcontrollers without a preceding modeling can be successful the same is not true if there is a need for reliable real-time systems with concurrent processes and communication. In these cases a formal representation of the system and its surrounding physical processes is necessary. To be able to apply these techniques students have to consider concepts of the other competence subdimensions on different cognitive levels (see Table 2).

Table 3: Example: Models

Profile	Students should be able to apply techniques of modeling like schematic diagrams and description languages to create models for a range of problem areas
AEs	Model and simulate a digital system using schematic diagrams. Create state and transition diagrams for simple problem domains Model and simulate a digital system using a hardware description language, such as VHDL or Verilog Demonstrate the ability to apply the techniques of modeling and simulation to a range of problem areas

In order not to go beyond the scope of this paper the competence profile of system modeling is considered only. As mentioned earlier, the author focuses on the first teaching at universities. Thus, not all types of models are of the same relevance and accessibility.

> ”In order for the composition to be well understood, we need first for the individual components to be well understood, and then for the meaning of the interaction between components to be well understood (Lee & Seshia, 2012, p. 132).”

This is in line with Marwedel (2011) who introduced finite state machines to model discrete dynamics and its timing extensions first. However, their combination to concurrent models has to be mediated later in the study. This is also true for hybrid systems that combine discrete and dynamic behavior. Such a distinction based on didactic metaknowledge exemplifies the proceeding and has to be done more in detail.

CONCLUSION

The need for research on embedded systems’ education was justified within the introduction. The main motivation is to support the first teaching at university where basic research about competences from the KOMINA project gets incorporated to a didactic system. This system with its cognitive structures, task classes, and learning aids was outlined. Here, the need for an adaptation of already existing didactic systems was justified with the subjects’ particularities and current research about competence orientation. Textbooks and curricula recommendations were analyzed in order to differentiate the competence dimension system design. System modeling as a key technique served as an example and was considered more in detail.

FURTHER WORK

The cognitive structures have to be explored and linked to the assumed competences as preconditions. This requires an analysis of necessary knowledge as well as adequate problem-solving strategies. Furthermore, the integration of modeling in the laboratory course will be performed.

REFERENCES

ACM/IEEE. (2004). *Curriculum guidelines for undergraduate degree programs in computer engineering: A report in the computing curricula series*. Retrieved from http://www.acm.org/education/education/curric_vols/CE-Final-Report.pdf

ACM/IEEE. (2005). *Computing curricula 2005: The overview report*. Retrieved from http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf

ACM/IEEE. (2008). *Computer science curriculum 2008: An interim revision of cs 2001*. Retreieved from http://www.acm.org/education/curricula/ComputerScience2008.pdf

Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., ... Wittrock, M. C. (2000). *A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives, abridged edition* (2nd ed.). Allyn & Bacon.

Artist Education Group. (2003). *Guidelines for a graduate curriculum on embedded software and system.* Retrieved from http://www.artist-embedded.org/docs/Publications/Education.pdf

Brinda, T. (2004). *Didaktisches system für objektorientiertes modellieren im informatikunterricht der sekundarstufe II*. Doctoral dissertation. Universität Siegen, Siegen, Germany.

Brinda, T., & Schubert, S. (2001). *Didactic system for object-oriented modelling*. In: Watson, Deryn ; Andersen, Jane (Ed.) : Networking the learner. Computers in education, ifip world conference on computers in education - wcce 2001. July 29 - August 03, 2001. Vol. 217, 473–482. Copenhagen, Denmark.

Caspi, P., Sangiovanni-Vincentelli, A., Almeida, L., Benveniste, A., Bouyssounouse, B., Buttazzo, G., ... Yi, W. (2005). *Guidelines for a graduate curriculum on embedded software and systems*. ACM Trans. Embed. Comput. Syst, 4(3), 587–611.

Freischlad, S. (2008). *Knowledge networks for internetworking in the process of course design.* In LYICT. Kuala Lumpur, Malaysia, July 7-10: IFIP.

Freischlad, S. (2010). *Entwicklung und erprobung des didaktischen systems internetworking im informatikunterricht*. Doctoral dissertation. Universität Siegen, Siegen, Germany.

Fuller, U., Riedesel, C., Thompson, E., Johnson, C. G., Ahoniemi, T., Cukierman, D., ... Thompson, D. M. (2007). *Developing a computer science-specific learning taxonomy*. ACM SIGCSE Bulletin, 39(4), 152-170.

GI (Ed.). (2011). *Curriculum technische informatik in bachelor- und masterstudiengängen informatik*. Bonn, Germany.

Jaschke, S., Büchner, S., Schubert, S., Schäfer, A., & Brück, R. ([in press]). *Competence oriented embedded systems course for computer science students.* In P. Marwedel, J. Jackson, & K. Ricks (Eds.), Workshop on embedded and cyber-physical systems education. Tampere, Finland.

Jaschke, S., Schubert, S., Schäfer, A., Brück, R., Kleinert, B., Schmidt, H., & Fey, D. (2011). *Competence research: Teaching embedded micro/nano systems*. In J. Jackson, P. Marwedel, & K. Ricks (Eds.), Workshop on embedded systems education. Taipei, Taiwan: ACM. doi:10.1145/2077370.2077373

Kleinert, B., Schmidt, H., Fey, D., Jascke, S., Büchner, S., Schubert, S., … Brück, R. (2012). *Approaches to teaching future computation technologies and nanosystems*: Proceedings of the ifip-conference. Retrieved from http://www.esri.mmu.ac.uk/aecrict

Lee, E. A., & Seshia, A. A. (2012). *Introduction to embedded systems, a cyber-physical systems approach*. UC Berkley, CA, USA.

Marwedel, P. (2005). *Towards laying common grounds for embedded system design education.* In the first workshop on embedded system education (WESE)

Marwedel, P. (2011). *Embedded system design: Embedded systems foundations of cyber- physical systems (2nd Etd)*. Springer Verlag.

Marwedel, P., & Engel, M. (2011). *Embedded system design 2.0: rationale behind a textbook revision.* In J. Jackson, P. Marwedel, & K. Ricks (Eds.), Workshop on embedded systems education. Taipei, Taiwan: ACM. doi:10.1145/2077370.2077372

Schäfer, A., Brück, R., Büchner, S., Jaschke, S., Schubert, S., Fey, D., ... Schmidt, H. (2012). *The empirically refined competence structure model for embedded micro- and nanosystems*: Proceedings of the 17th acm annual conference on innovation and technology in computer science education, 57–62. New York, NY, USA: ACM. doi:10.1145/2325296.2325314

Schubert, S., & Stechert, P. (2010). *Competence model research on informatics system application*: Proceedings of the ifip conference new developments in ict and education. Amiens, France, June 28-30.

Weinert, F. E. (2001). *Leistungsmessungen in schulen.* Beltz Pädagogik.

Biography

Steffen Jaschke received his diploma in Computer Science (2010) from the University Siegen, Germany. Since October 2010 he is employed as a research assistant at the University of Siegen. His main research interests include education methodologies for embedded systems.