International Journal of Research Studies in Computing
2015 April, Volume 4 Number 1, 3-14

Process-related competence areas to computer science education: An empirical determination

Seitz, Cornelia ✉
University of Education Ludwigsburg, Germany (seitz@ph-ludwigsburg.de)
Zendler, Andreas
University of Education Ludwigsburg, Germany (zendler@ph-ludwigsburg.de)

ISSN: 2243-772X
Online ISSN: 2243-7797

OPEN ACCESS

Received: 8 October 2014 **Revised**: 1 November 2014 **Accepted**: 8 November 2014
Available Online: 1 December 2014 **DOI:** 10.5861/ijrsc.2014.932

Abstract

In this article empirically determined, process-related competence areas to computer science education at primary and secondary school levels are discussed. Based on the results of interviews with 120 professors of computer science and using multi-dimensional scaling and cluster analysis, six process-related competence areas to computer science education characterized by the degree of content-related coverage and educational accessibility can be defined: critical thinking, ordering, abstracting, problem solving, analyzing and collaborative construction. These competence areas consist of central process concepts of computer science (e.g., classifying, problem solving and posing) combined with central content concepts of computer science (e.g., algorithm, system, process).

Keywords: computer science education; competence model; process concepts; content concepts; competence-based education

Process-related competence areas to computer science education: An empirical determination

1. Introduction

Already in the mid-1980s competence-oriented approaches in education were initiated by discussions in Anglo-American educational commissions (Ravitch, 1995; Horn 2004; Klieme & Maag Merki, 2008). Since then numerous approaches have been developed favouring different concepts of competence (Rychen & Salganik, 2003a; Klieme, Hartig, & Rauch, 2008). An overview is provided by Ennis (2008). The focus in competence-oriented approaches is on output as results of task-solving (performance) by learners, on several knowledge components (knowledge, skills, dispositions, attitudes), on the acquisition of competences and on standardized methods of competence assessment.

The competence concept gained a high degree of international significance in the context of the performance studies among school students within PISA (Programme for International Student Assessment) under the umbrella of OECD (Organisation for Economic Cooperation and Development). In the DeSeCo (Defining and Selecting Key Competencies) project, the OECD defines competence as follows: “A competence is defined as the ability to successfully meet complex demands in a particular context. Competent performance or effective action implies the mobilization of knowledge, cognitive and practical skills, as well as social and behaviour components such as attitudes, emotions, and values and motivations. A competence is therefore not reducible to its cognitive dimension, and thus the terms competence and skill are not synonymous.” (Rychen & Salganik, 2003b, p. 2)

No complete competence model has been developed for computer science education so far. Merely individual parts of a complete model have been considered. It is agreed that the dimensions of knowledge, skills and attitudes are important for defining competence (e.g., Sakhnini & Hazzan, 2008; Crescenzi, Loreti, & Pugliese, 2006; Lancaster & Lancaster, 2006; Fuller & Keim, 2008). The competence dimension skills covers sensomotoric aspects of individual performance, e.g. craft and technical skill, manual dexterity or language ability (Gnahs, 2007). In computer science education these skills are relevant for programming, modelling, communication and cooperation (Crescenci, Loreti, Pugliese, 2006; Liew, 2005). The competence dimension disposition refers to personality traits like openness, conscientiousness and agreeableness (Costa & McCrae, 1997; Gnahs, 2007; Lindenberger & Schaefer, 2008). Dispositions relevant in computer science education become obvious in questions of researching and exploring (Katz, 1993; Lancaster & Lancaster, 2006). The competence dimension attitudes relates to values and beliefs people have acquired towards objects, ideas, other persons or behaviours (Schmuck & Kruse, 2005; Damon, Lerner, & Eisenberg, 2006; Gnahs, 2007; Montada, 2008). The need to discuss values and attitudes in computer science education underlines the Final Report of the ACM K- 12 Task Force Curriculum Committee: “A graduate should (…) understand the social, professional, and ethical issues involved in the use of computer technology.” (Tucker et al., 2003, p. 4-5). Examples for attitudes that could be discussed in computer science education are quality, trustworthiness, honesty and collegiality.

Competence areas define the subset of concepts of a subject area on which a competence model for the subject-related domain builds. Competence areas for computer science education characterized by content concepts (e.g., algorithm, system, process) rather than process concepts (e.g., analyzing, classifying, problem solving and posing) are described by CSTA (Tucker et al., 2006). Competence areas for computer science education of which content and process concepts are typical were developed by the Kultusministerkonferenz (German Education Ministers’ Conference) (KMK, 2007) and the Gesellschaft für Informatik (German Society for Computer Science) (GI, 2008). However, these concepts are based on educational reflections by individual authors; they are not substantiated empirically (Furst, Isbell, & Guzdial, 2007).

This point is addressed by the authors of this study. The target is to determine process-related competence areas to computer science education on an empirical basis. The study is organized as follows: First the methods and the general procedure applied for the data analyses are described. Then a detailed description of the results is given and finally the results are discussed.

2. Method

2.1 Starting points for determining competence areas

The determination of competence areas for computer science education is based on the results of the taxonomies of content and process concepts in computer science (Zendler & Spannagel, 2008; Zendler, Spannagel, & Klaudt, 2008, Zendler, Spannagel, & Klaudt, 2011; Zendler, Klaudt, Spannagel, & Reuter, 2013; Zendler & Hubwieser, 2013).

2.2 Central content concepts

The first study (Zendler & Spannagel, 2008) was conducted under the aspect of the structure of the discipline approach. 37 professors of computer science assessed 49 content concepts of computer science that had previously been selected by a frequency analysis of the ACM Computing Classification System (1998 Version). The concepts were assessed by horizontal, vertical, time and sense criteria (cf. Schwill, 1994). Via cluster analysis the following 15 central content concepts were determined: algorithm, computer, data, problem, information, system, language, program, test, communication, software, process, model, computation and structure.

2.3 Central process concepts

The second study (Zendler, Spannagel, & Klaudt, 2008) was conducted on the background of process as content (Parker und Rubin, 1966; Costa & Liebmann, 1997a; 1997b; 1997c) approach. Applying the same criteria as in the first study, 24 professors of computer science assessed 44 process concepts. Data of the second study was analyzed in the same manner as the first study data. The process concepts were selected from the list by Costa and Liebmann (1997d). The following 16 central process concepts were determined: problem solving and problem posing, analyzing, generalizing, finding relationships, classifying, investigating, ordering, communicating, presenting, categorizing, finding cause-and-effect relationships, comparing, collaborating, creating and inventing, transferring und questioning.

2.4 Data basis

As a data basis the data matrix from the study by Zendler, Spannagel and Klaudt (2008) is used (Figure 1). In that study, the professors of computer science assessed the importance of process concepts for dealing with the content concepts on a scale from 0 (“no importance“) to 5 (“very high importance“). The data matrix contains the mean values of all content concept/process concept combinations. The rows and columns are sorted by decreasing grand means.

2.5 Data analysis

Data is analyzed as follows: The data matrix (see Figure 1) with the content concept means in relation to the process concepts serves as data basis.

In a first step, multi-dimensional scaling (MDS) using the ALSCAL algorithm (Takane, Young, & de Leeuw, 1977; Young, Takane, & Lewyckyij, 1978) is performed. For this, a distance matrix is developed for the process concepts, for which the content concepts serve as variables. They establish a 15-dimensional space in which the 16 process concepts are placed. The Euclidian distance is used as distance measure. The method of multi-dimensional scaling produces an MDS configuration of the process concepts.

In a second step, a cluster analysis is performed for the MDS configuration of the process concepts. The authors applied Ward's method, a hierarchical procedure that calculates a series of cluster partitions (Everitt, Landau, & Leese, 2001). The Euclidian distance served as distance measure.

Multi-dimensional scaling and cluster analysis were calculated by SPSS 20.0.

process concepts \ **content concepts**	problem	information	model	algorithm	data	structure	system	computation	process	software	program	test	communication	language	computer	***grand means***
analyzing	4.54	4.29	4.17	4.46	4.00	4.29	4.25	3.79	4.13	4.04	4.08	3.29	3.38	3.46	2.83	**3.93**
classifying	3.83	4.21	3.38	3.42	3.83	3.33	2.79	3.00	2.96	2.79	2.63	2.38	2.42	2.92	2.58	**3.10**
problem solving and posing	4.50	2.63	3.46	4.25	2.50	2.33	3.17	3.71	2.63	3.17	3.50	2.71	2.25	2.83	2.25	**3.06**
categorizing	3.63	3.79	3.42	3.04	3.75	3.08	2.75	2.67	3.04	2.71	2.38	2.63	2.29	3.08	2.92	**3.01**
investigating	4.00	2.83	3.29	3.58	2.83	3.13	3.17	3.08	3.04	2.67	2.71	3.04	2.71	2.42	2.21	**2.98**
finding relationships	3.08	3.79	3.46	2.54	3.71	3.50	2.92	2.92	2.71	2.29	2.21	2.79	2.63	2.67	1.67	**2.86**
generalizing	3.17	3.42	4.13	3.83	2.92	3.79	2.79	3.04	2.42	2.50	2.51	1.92	2.17	2.71	1.50	**2.85**
creating and inventing	2.04	1.83	3.58	4.13	1.50	2.67	2.83	2.67	2.00	3.67	3.38	3.00	2.25	2.67	2.50	**2.71**
comparing	2.88	3.25	3.13	3.46	3.25	2.42	2.04	2.75	2.42	2.08	2.13	2.88	2.08	2.33	2.21	**2.62**
finding cause-and-effect r.	2.88	2.50	2.75	2.67	2.79	2.75	2.63	2.71	2.92	2.38	2.33	3.63	2.58	1.75	1.96	**2.62**
questioning	3.54	3.00	2.63	2.38	2.75	2.58	2.46	2.46	2.38	2.25	2.04	3.21	2.63	1.88	2.29	**2.61**
transferring	2.83	2.92	3.13	3.33	2.33	2.71	2.46	2.46	2.33	2.46	2.75	2.08	2.13	2.13	1.96	**2.56**
communicating	2.50	2.58	2.38	1.96	2.17	1.54	2.46	2.08	3.00	2.79	2.38	1.83	4.13	2.88	2.88	**2.53**
presenting	2.50	3.54	2.58	2.46	3.42	2.88	2.79	2.42	2.13	2.17	2.25	2.17	2.33	2.08	1.79	**2.50**
collaborating	2.58	1.96	2.46	1.42	1.54	1.42	2.71	2.13	2.67	3.00	2.42	1.96	3.50	2.17	2.83	**2.32**
ordering	1.75	3.33	1.88	1.67	3.71	2.17	1.38	1.58	1.71	1.13	1.42	1.42	1.38	1.33	1.04	**1.79**
grand means	**3.14**	**3.12**	**3.11**	**3.04**	**2.94**	**2.79**	**2.72**	**2.72**	**2.65**	**2.63**	**2.57**	**2.56**	**2.55**	**2.46**	**2.21**	

Figure 1. Means of process and content concepts (N=24) – sorted by grand means

3. Results

3.1 MDS configuration of the process concepts

Multi-dimensional scaling provides an MDS configuration of the concepts with dimensions that can be interpreted semantically. As a method, multi-dimensional scaling is applied as structural analysis to graphically highlight similarities/dissimilarities among concepts (see Borg & Groenen, 2005; Borg & Staufenbiel, 2007).

Figure 2 shows the resulting MDS configuration of the process concepts in a 2-dimensional coordinate system. It shows that six different regions can be identified. The first region containing the five process concepts presenting, comparing, questioning, finding cause-and-effect-relationships and transferring is characterized by average values in both dimensions. The second region with ordering is characterized by low values in both dimensions. The third region with the four concepts generalizing, categorizing, classifying, investigating and finding relationships is characterized by average values in dimension #1 and rather low values in dimension #2. The forth region containing investigating and problem solving and posing shows average values in dimension #1 and high values in dimension #2. The fifth region containing analyzing shows high values in dimension #1 and average values in dimension #2. The sixth regions containing the concepts creating and investigating, collaborating and communicating shows average values in dimension #1 and high values in dimension #2.

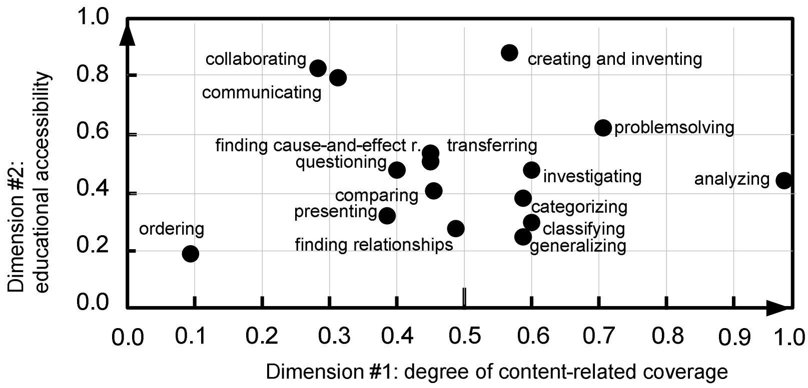

Figure 2. MDS configuration of process concepts

The MDS configuration was assessed by Young's S-stress formula 1 and the squared correlation of the distances with disparities R2 (see Borg & Groenen, 2005). The following is obtained for the MDS configuration: S-stress 1= .12 and R2 = .95. Thus, the adaptation of the MDS configuration with dimensionality 2 to the raw data can be assessed as being acceptable after a comparison with the simulation results from Borg and Groenen (2005, chapter 3).

Dimension #1. This dimension can be interpreted as degree of content-related coverage: Process concepts with high values (> 2.50) for many content concepts have a high degree of content-related coverage (e.g., analyzing, classifying, investigating). Process concepts with low values for content concepts have a low degree of content-related coverage (e.g., communicating, collaborating, ordering). A process concept (see Figure 2) with very high degree of process-related coverage is analyzing, for which the following 15 content concepts are relevant: problem, information, model, algorithm, data, structure, system, computation, process, software, program, test, communication, language, and computer. A process concept with a very low degree of content-related coverage is ordering, for which only two content concepts are significant: information and data.

Dimension #2. This dimension can be interpreted as educational accessibility and characterizes process concepts whether they build on other concepts, whether their introduction is linked to certain resources (hardware or software) or whether teaching presupposes a certain intellectual level. A process concept with easy educational accessibility, for example, is ordering. This concept can be introduced in computer science education without reference to other process concepts of computer science. Besides, it can be introduced without access to other resources and it can be taught on virtually any intellectual level. Examples of process concept with less easy educational accessibility are creating and inventing or transferring. Their introduction in computer science education depends on the prior introduction of other computer science concepts (such as questioning, comparing and classifying). Besides, some of them can only be dealt with if appropriate resources (software, hardware) are available. And these process concepts cannot easily be introduced on any intellectual level (Zendler & Spannagel, 2008).

3.2 Competence areas determined

The MDS configuration indicates that the process concepts can be represented with respect to two semantic dimensions, i.e., the degree of content-related coverage and the educational accessibility. Competence areas for computer science education are now identified by cluster analysis. As a starting point the coordinates of the process concepts in the MDS configuration are used (see Figure 2): degree of content-related coverage and educational accessibility.

The results of the cluster analysis can be seen in Figure 3: Six clusters are obtained for the 16 process concepts. The C-index of Hubert and Levin (1976) was considered for selecting the stop criterion for determining the number of clusters ("cut" in the illustration below). In the following, the clusters will be presented as competence areas.

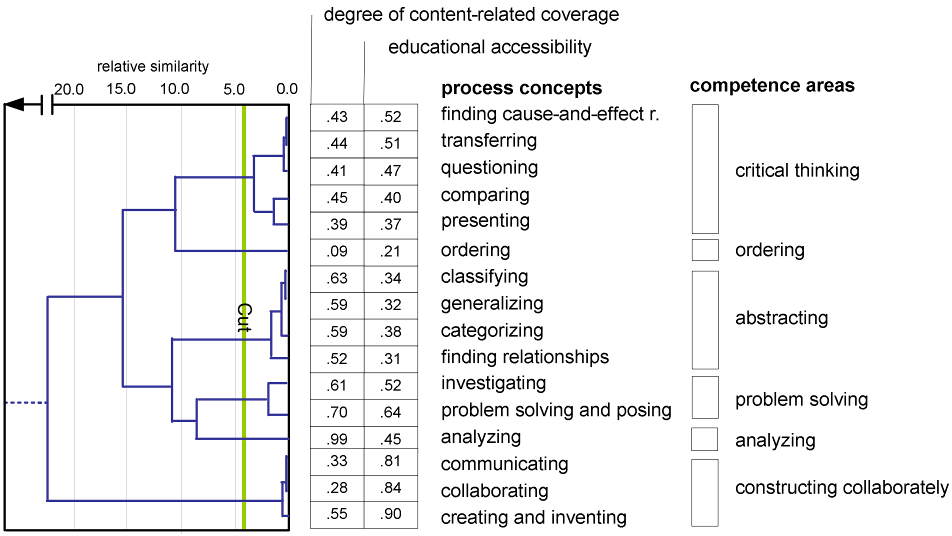

Figure 3. Assignment of process concepts to competence areas

Critical Thinking. This competence comprises the five process concepts finding relationships, presenting, comparing, questioning, finding cause-and-effect-relationships and transferring which are merged early due to their similar values in relation to the degree of content-related coverage and educational accessibility. Typical of this competence area is an average educational accessibility and a rather low degree of content-related coverage.

Ordering. This competence area consists only of the process concept ordering. It has a low degree of content-related coverage but is very easily accessed educationally.

Abstracting. This competence area contains the four process concepts classifying, generalizing, categorizing und finding relationships. Typical of this competence area is an average degree of content-related coverage and an easy educational accessibility.

Problem solving. This competence area comprises the two concepts investigating and problem solving and posing. It is characterized by its high content-related coverage and rather low educational accessibility.

Analyzing. This competence area consists of only the concept analysing. I has a very high content-related coverage (analyzing can be combined with almost every content concept), but the educational accessibility is average.

Collaborative construction. This competence area contains the three process concepts creating and inventing, communicating and collaborating. It is characterized by low values in educational accessibility while their content-related coverage differs.

4. Discussion

This study makes an important contribution for the development of an empirically validated competence model to computer science education. It was possible to determine the competence areas empirically whose process concepts play a central role in many areas of computer science, which can be taught at different intellectual levels, are of longer-term relevance and are related to language and/or thinking in everyday life and world in which we live.

Because of their easy educational accessibility the ordering and the critical thinking competence areas are suited particularly for initial education in computer science. The competence area abstracting shows an average level of both educational accessibility and content-related coverage and can be taught in an advanced computer science course. The problem solving and analyzing competence areas with the high level of content-related coverage of its process concepts should be preferred for computer science education which comprises all competence levels. The collaborative construction competence area is suitable for team-oriented computer science courses.

The results of this study support the relevant literature, particularly concerning problem solving and algorithmic thinking (Fadi & McHugh, 2001, Muller & Haberman, 2008), system-level-perspective (Gal-Ezer & Zeldes, 2000; Ricks & Jackson, 2009), modelling (Mühling, Hubwieser, & Brinda 2010). They can be used for consolidating available curricular drafts for computer science as a teaching subject at school of the type available in many countries or currently in the state of revision, e.g., for England and Wales, Scotland, the USA, Israel, New Zealand, Germany, India, South Korea, Greece. (cf. Jones, 2011; Informatics Europe & ACM Europe Working Group on Informatics Education, 2013). In particular, the results can be regarded as empirical finding for current demands for the necessity of computer science education (Ericson et al., 2008; Jones, 2011) and can also be applied in the context of the research-based training of computer science teachers (EC, 2010; Ericson et al., 2008; Kansanen, 2006) in which the curriculum should comprise: development of computer science concepts, definition of computer science levels of competence, acquisition of competences in computer science, and methods for the assessment of competences in computer science.

Future research should focus on a more detailed specification of the process concepts for each competence area, also integrating the content concepts for computer science education from a didactic angle. The specification can be based on topic maps (Maicher, Sigel, & Garshol, 2007), in which relations between content and process concepts are defined more closely by relations of equivalence, by associative and hierarchic relations. Also, the competence areas themselves should be coordinated via topic maps. Furthermore, prototypical didactic examples that would lend themselves to didactic validation should be developed for all competence areas. The question of what methods and arrangements should be applied in teaching the content and process concepts can be ascertained from expert interviews beforehand.

5. References

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling*. Berlin: Springer.

Borg, I., & Staufenbiel, T. (2007). *Textbook of theories and methods of scaling* [In German: *Lehrbuch Theorien und Methoden der Skalierung*]. Bern: Huber.

Cassel, L. N., Davies, G., Finley, G. T., Hacquebard, A., LeBlanc, R., Mann, S., McGettrick, A., Riedesel, C., Sloan, R. H., & Varol, Y. L. (2005). A synthesis of computing concepts. *ACM SIGCSE Bulletin, 37*(4), 162–172. http://dx.doi.org/10.1145/1113847.1113894

Cassel, L. N., Davies, G., Fone, W., Hacquebard, A., Impagliazzo, J., LeBlanc, R., Little, J. C., McGettrick, A., & Pedrona, M. (2007). The computing ontology - application in education. *ACM SIGCSE Bulletin, 39*(4), 171–183. http://dx.doi.org/10.1145/1345375.1345439

Costa, A. L., & Liebmann, R. M. (Eds.). (1997a). *Envisioning process as content. Toward a renaissance curriculum*. Thousand Oaks, CA: Corwin Press.

Costa, A. L., & Liebmann, R. M. (Eds.). (1997b). *The process-centered school. Sustaining a renaissance community*. Thousand Oaks, CA: Corwin Press.

Costa, A. L., & Liebmann, R. M. (Eds.). (1997c). *Supporting the spirit of learning. When process is content*. Thousand Oaks, CA: Corwin Press.

Costa, A. L., & Liebmann, R. M. (1997d). Toward renaissance curriculum. In L. A. Costa & R. M. Liebmann (Eds.), *Envisioning process as content. Toward a renaissance curriculum* (pp. 1–20). Thousand Oaks, CA: Corwin Press.

Costa, P. T., & McCrae, R. R. (1997). Longitudinal stability of adult personality. In R. Hogan, J. Johnson, & S.

R. Briggs (Eds.), *Handbook of personality psychology* (pp. 269–290). Orlando, FL. Academic Press. http://dx.doi.org/10.1016/B978-012134645-4/50012-3

Crescenzi, P., Loreti, M., & Pugliese, R. (2006). Assessing CS1 Java skills: a three-year experience. In *Proceedings of the 11th annual SIGCSE conference on innovation and technology in computer science education – ITiCSE06* (p. 348). New York: ACM.

Damon, W., Lerner, R., & Eisenberg, N. (Eds.) (2006). *Handbook of child psychology 3: social, emotional, and personality development: 3 (Handbook of child psychology)*. New York: Wiley.

Ennis, M. R. (2008). *Competency models: a review of the literature and the role of the employment and training administration*. Washington, DC: U. S. Department of Labor.

Ericson, B., Armoni, M., Gal-Ezer, J., Seehorn, D., Stephenson, C., & Trees, F. (2008). Ensuring exemplary teaching in an essential discipline: Addressing the crisis in computer science teacher certification. *Final Report of the CSTA Teacher Certification Task Force. ACM*.

European Commission (EC). (2010). Common European principles for teacher competences and qualifications. Retrieved November 7, 2013 from http://ec.europa.eu/education/policies/2010/doc/principles_en.pdf

Everitt, B. S., Landau, S., & Leese, M. (2001). *Cluster analysis*. London: Arnold.

Fadi, P. D., & McHugh, J. A. (2000). Prototype software development tools for beginning programming. *Journal of Computer Science Education, 14*(3&4), 14–20.

Fuller, U., & Keim, B. (2008). Assessing students' practice of professional values. In *Proceedings of the 13th Annual SIGCSE conference on innovation and technology in computer science education – ITiCSE08* (pp. 88–92). New York: ACM.

Furst, M., Isbell, C., & Guzdial, M. (2007). Threads™: How to restructure a computer science curriculum for a flat world. *ACM SIGCSE Bulletin, 39*(1), 420–424. http://dx.doi.org/10.1145/1227504.1227456

Gal-Ezer, J., & Zeldes, A. (2000). Teaching software designing skills. *Computer Science Education, 10*(1), 25–38. http://dx.doi.org/10.1076/0899-3408(200004)10:1;1-P;FT025

GI. Gesellschaft für Informatik (2008). Principles and standards for computer science in school [In German: Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I]. *LOG IN, 28* (150/151) supplement.

Gnahs, D. (2007). *Competences – acquisition, measurement, instruments* [In German*: Kompetenzen – Erwerb, Erfassung, Instrumente*]. Bielefeld: Bertelsmann.

Horn, R. A. J. (2004). *Standards*. New York: Lang.

Hubert, L. J., & Levin, J. R. (1976). A general statistical framework for assessing categorical clustering in free recall. *Psychological Bulletin, 83*, 1072–1080. http://dx.doi.org/10.1037/0033-2909.83.6.1072

Informatics Europe & ACM Europe Working Group on Informatics Education, (2013). *Informatics education: Europe cannot afford to miss the boat.* Retrieved November 7, 2013, from http://germany.acm.org/upload/pdf/ACMandIE report.pdf

Jones, S.P. (2011). *Computing at School. International comparisons*. Retrieved May 7, 2013, from http://csta.acm.org/About/sub/AboutFiles/IntlComparisons v5.pdf

Kansanen, P. (2006). Constructing a research-based program in teacher education. In F. Oser, F. Achtenhagen, & U. Renold (Eds.), *Competence Oriented Teacher Training* (pp. 11-22). Rotterdam: Sense Publishers.

Katz, L. G. (1993). *Dispositions as educational goals*. Urbana, IL: ERIC Clearinghouse on Elementary and Early Childhood Education.

Klieme, E., Hartig, J., & Rauch, D. (2008). The concept of competence in educational contexts. In J. Hartig, E. Klieme, & D. Leutner (Eds.), *Assessment of competencies in educational contexts* (pp. 3–22). Göttingen: Hogrefe.

Klieme, E., & Maag Merki, K. (2008). Introduction of educational standards in German-speaking countries. In J. Hartig, E. Klieme, & D. Leutner (Eds.), *Assessment of competencies in educational contexts* (pp. 305–314). Göttingen: Hogrefe.

KMK. Kultusministerkonferenz (2007). *Uniform requirements in informatics* [In German: *Einheitliche Prüfungsanforderungen Informatik*]. Retrieved from http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01_EPA_Informatik.p df

Lancaster, S. J. C., & Lancaster, P. E. (2006). Technology shaping learner dispositions. *The Knowledge Tree: An e-journal of learning innovation, 12*, 8–19.

Liew, C. W. (2005). Teaching software development skills early in the curriculum through software engineering. In *Proceedings of the 10th Annual SIGCSE conference on innovation and technology in computer science education – ITiCSE05* (pp. 133–137). New York: ACM.

Lindenberger, U., & Schaefer, S. (2008). Adulthood and age [In German: Erwachsenenalter und Alter]. In R. Oerter & L. Montada (Eds.), *Entwicklungspsychologie* (pp. 366–409). Weinheim. Beltz.

Maicher, L., Sigel, A., & Garshol, L. M. (2007). *Leveraging the semantics of topic maps*. Berlin: Springer. http://dx.doi.org/10.1007/978-3-540-71945-8

Montada, L. (2008). Moral development and socialisation [In German: Moralische Entwicklung und Sozialisation]. In R. Oerter & L. Montada (Eds.), *Entwicklungspsychologie* (pp. 572–606). Weinheim: Beltz.

Mühling, A., Hubwieser, P., & Brinda, T. (2010). Exploring teachers' attitudes towards object oriented modelling and programming in secondary schools. In ACM (Ed.), *ICER '10: Proceedings of the Sixth international workshop on computing education research* (pp. 59–68). New York, NY: ACM.

Muller, O., & Haberman, B. (2008). Supporting abstraction processes in problem solving through pattern-oriented instruction. *Computer Science Education, 18*(3), 187–212. http://dx.doi.org/10.1080/08993400802332548

Parker, J.C., & Rubin, L. J. (1966). Process as content. Curriculum design and the application of knowledge. Chicago: Rand McNally & Company.

Ravitch, D. (1995). *National standards in American education*. Washington, D.C.: Brookings Institution Press.

Ricks, K. G., & Jackson, D. J. (2009). Incorporating system-level concepts into undergraduate embedded systems curricula. *ACM SIGBED Review 6*(1), Article No. 1. http://dx.doi.org/10.1145/1534480.1534481

Rychen, S., & Salganik, L. H. (2003a). Key competencies for a successful life and well-functioning society. Göttingen: Hogrefe.

Rychen, S., & Salganik, L. H. (2003b). Definition and selection of competencies: theoretical and conceptual foundations – summary of the final report "key competencies for a successful life and a well-functioning society. Retrieved from http://www.oecd.org/education/skills-beyond-school/41529556.pdf

Sakhnini, V., & Hazzan, O. (2008). Reducing abstraction in high school computer science education: the case of definition, implementation, and use of abstract data types. *ACM Journal of Educational Resources in Computing, 8*(2), Article No. 5. http://dx.doi.org/10.1145/1362787.1362789

Schmuck, P., & Kruse, A. (2005). Development of values and life goals [In German: Entwicklung von Werthaltungen und Lebenszielen]. In J. Asendorpf (Ed.), *Enzyklopädie der Psychologie: Soziale, emotionale und Persönlichkeitsentwicklung* (pp. 191–258). Göttingen: Hogrefe.

Schwill, A. (1994). Fundamental ideas of computer science. *EATCS Bulletin, 53*, 274–295.

Takane, Y., Young, F. W., & de Leeuw, J. (1977). Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. *Psychometrika, 42*, 8–67. http://dx.doi.org/10.1007/BF02293745

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). *A model curriculum for K-12 computer science*. New York: ACM.

Young, F. W. Takane, Y., & Lewyckyij, R. (1978). Three notes on ALSCAL. *Psychometrika, 43*, 433–435.

Zendler, A., & Spannagel, C. (2008). Empirical foundation of central concepts for computer science education. *ACM Journal on Educational Resources in Computing, 8*(2), Article No. 6. http://dx.doi.org/10.1145/1362787.1362790

Zendler, A., Spannagel, C., & Klaudt, D. (2008). Process as content in computer science education: empirical determination of central processes. *Computer Science Education, 18*(4), 231–245. http://dx.doi.org/10.1080/08993400802390553

Zendler, A., Spannagel, C., & Klaudt, D. (2011). Marrying content and process in computer science education. *IEEE Transactions on Education, 54*(3), 387–397. http://dx.doi.org/10.1109/TE.2010.2062184

Zendler, A., & Hubwieser, P. (2013). The influence of teacher training programs on evaluations of central computer science concepts. *Teacher and Teaching Education, 34*, 130–142. http://dx.doi.org/10.1016/j.tate.2013.03.005

Zendler, A., Klaudt, D., Spannagel, C., & Reuter, T. (2013). Semantic categorization of content and process concepts relevant to computer science education. *International Journal of Research Studies in Computing, 2*(1), 3–10.

Appendix: Definitions of content and process concepts

Content concepts

- *Algorithm*. An algorithm is a list of instructions to solve a class of problems.
- *Communication*. (Computer based) communication is understood as any form of human interaction by using two or several computers in a network.
- *Computation*. Computation is understood as a process that as a model is expressed in an algorithm.
- *Computer*. A computer is a programmable machine to carry out a sequence of operations.
- *Data*. Data are signs to represent facts according to syntactic rules in a sign (system).
- *Information*. Information is understood as data that has meaning according to contextual relations.
- *Language*. Language is understood as a set of words which are produced according to semiotic rules (syntax, semantic) in a given alphabet.
- *Model.* A model is a system (isomorphic) mapping elements of a domain to elements of a range with statements for purpose and usage.
- *Problem*. A problem is a task whose solution is to transform an unsatisfactorily current state to a satisfactorily goal state.
- *Program*. A (computer) program is a sequence of instructions performing a task.
- *Process.* A process transforms inputs into outputs by using controls and mechanisms (methods and resources).
- *Software*. Software is understood as set of programs that are performed by a computer.
- *Structure*. Structure is understood as the entirety of the elements of an object (system, data, program ...), its relations and its functions.
- *System.* Systems are complex entities that (1) consist of components (or subsystems), that (2) are delimited against an environment, and (3) their components have relations among one another and the environment.
- *Test.* A (software) test is a process to verify and/or to validate software.

Process concepts (from Costa & Liebmann, 1997a, Chapter 1)

- *Analyzing*. Separating or breaking up a whole into its parts according to some plan or reason.
- *Categorizing*. Arranging items that possess the same properties according to a predetermined scheme, criterion, or rule.
- *Classifying*. Adding ideas or objects to a group according to an established or invented scheme, criterion, or rule.
- *Collaborating*. Knowing how to reciprocate and work with others on a common endeavour in ways that are mutually productive by aligning personal goals with larger group goals, seeking consensus, resolving conflicts and disagreements with dignity and respect, valuing and protecting the common

good, volunteering, and being of service to and feeling compassion for others.

- *Communicating*. Engaging in productive discussions, discourse, debate, and dialogue in their multiple forms (linguistic, visual, technological, kinaesthetic, musical, etc.) by including such skills as listening with understanding and empathy; speaking and writing with clarity; using precision and accuracy; and knowing how to behave when value, language, belief, cultural, gender, or religious differences may cause misunderstanding, confusion, or difficulties in interpersonal interactions.

- *Comparing*. Examining the attributes of two or more objects to discover resemblances or similarities.

- *Creating and inventing*. Generating new ideas and products of interest and value to others; accepting that each human being has a large capacity and potential for creative thought and action; learning how to access inner resources to cause the creative juices to flow when a situation demands it.

- *Finding cause-and-effect relationships*. Investigating and stating the reasons for a set of results that is produced.

- *Finding relationships*. Detecting regularity between two or more operations: temporal, causal, syllogistic, transitive, spatial, mathematical, and so forth.

- *Generalizing*. Formulating a rule, principle, or formula that governs or explains a number of related situations.

- *Investigating*. Knowing how to persist and conduct extended research to a satisfactory completion; drawing on multiple resources, organizing and synthesizing into a coherent whole, and presenting the results in a succinct manner.

- *Ordering*. Arranging conditions, objects, events, and ideas according to an established schema or criterion in relationship to one another.

- *Presenting*. Communicating knowledge of concepts, principles, and skills in a coherent manner to a particular audience in a manner that produces and enhances the audience's understanding.

- *Problem solving and problem posing*. Being aware that a problem exists; defining, refining, and clarifying a problematic situation; analyzing a situation, describing the inherent problems in that situation, and designing and conducting a plan of action to resolve some discrepancy between an existing state and a desired state.

- *Questioning*. Making relevant inquiries to find information and to fill in the gaps between what is known and what is not known.

- *Transferring*. Applying what was learned in one situation to other similar, as well as to distant, dissimilar situations; finding opportunities to use what is known; drawing from previous knowledge and applying it to new and novel situations.