



Educational measures to computer science concepts: Degree of process-related coverage and degree of educational accessibility

Zendler, Andreas 

University of Education Ludwigsburg, Germany (zendler@ph-ludwigsburg.de)

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

Received: 23 February 2016
Available Online: 30 June 2016

Revised: 1 March 2016
DOI: [10.5861/ijrsc.2016.1469](https://doi.org/10.5861/ijrsc.2016.1469)

Accepted: 27 March 2016

OPEN ACCESS

Abstract

This article presents two measures to assess computer science content concepts from an educational point of view. The two measures are the degree of process-related coverage and the degree of educational accessibility. While the degree of process-related coverage determines the significance of process concepts for content concepts, the degree of educational accessibility can be understood as the degree to which content concepts are built upon other content concepts. The application of the two measures demonstrates how computer science content concepts can be semantically positioned so that conclusions can be drawn for their application in competence-oriented computer science education.

Keywords: computer science education; content concepts; process concepts; degree of educational accessibility; degree of process-related coverage

Educational measures to computer science concepts: Degree of process-related coverage and degree of educational accessibility

1. Introduction

The significance of computer science for the economy and for society in general is undisputed (Tucker & Wegner, 2009; Johnson & Miller, 2009). The *Towards 2020 Science* report (Emmott, 2006) highlights the growing importance of computer science at the interface with other branches of science and emphasizes the critical role of computer science concepts such as algorithm, process, and program/data across scientific fields: “Several fundamental computer science concepts are already on their way to becoming household names in science, and many more will follow” (Shapiro, Harel, Bishop, & Muggleton, 2006, p. 24). The authors illustrate the role of computer science concepts by reference to the example of the pair of concepts *program* and *data*: “(...) we expect core computer science concepts on interchangeability of program and data (...) to prove essential for the full understanding of the role of DNA as program and data” (Shapiro et al., 2006, p. 25).

Ericson (2008) identifies five reasons why computer science is becoming increasingly important in schools: (1) computer science leads to multiple career paths, (2) computer science is important intellectually across a variety of disciplines, (3) computer science is important to industry, (4) computer science supports and links to other sciences, and (5) computer science teaches both scientific and societal problem solving.

In curriculum planning of computer science education, the specification of learning contents is very important. This can be done using science-oriented approaches. The core idea behind science-oriented approaches is to gain learning contents from scientific disciplines based on scientific criteria (Zeichner, 1983; Driel van, Beijard, & Verloop, 2001; Zeidler, 2002; Fishman & Davis, 2006; Davis, 2008). This can be justified with three arguments: (1) science characterizes our insights of the world, (2) science provides structures to organize experiences, and (3) science provides a continuum of knowledge among laymen and experts.

In the literature, two basic types of science-oriented approaches are discussed: on the one hand the *structure of the discipline* approach by Bruner (1960), on the other hand the *process as content* approach developed by Parker and Rubin (1966), currently getting a renaissance through the trilogy of Costa and Liebmann (1997a, 1997b, 1997c). The first approach derives learning contents from fundamental ideas or basic concepts of the discipline, the second from methods used by scientists of the discipline.

The *structure of the discipline* approach is used in the context of *fundamental ideas* (Schwill, 1994; Wursthorn, 2005; Armoni & Ginat, 2008), *threshold concepts* (Eckerdal et al., 2006), and empirically determined *central content concepts* of computer science (Zendler & Spannagel, 2008). It is a starting point for curricular improvements (Franklin & Johnson, 2008). Modern *process as content* approaches are characterized by stressing four principles (Bolhuis, 2003): (1) self-directed learning, (2) focus on knowledge building in the domain, (3) emotional aspects of learning, and (4) learning processes as social phenomena. In the context of computer science education *central process concepts* empirically determined are also available (Zendler, Spannagel, & Klaudt, 2009). For the specific use of central concepts, the question arises to measures that allow statements which concepts can be used for consolidating available curricula, can be applied in the context of the research-based training, equip students with competences that will remain relevant in the longer term, and that are to some extent representative of the subject. So far, there are no such measures. This study seeks to provide a contribution to closing this gap.

The study is organized as follows: Section 2 addresses the methods and the general procedure applied for the data analysis of content. Section 3 contains a detailed description of the results. Section 4 discusses the results.

2. Method

2.1 Starting points for determining educational measures

The empirical determination of the educational measures for computer science education builds on the results of the taxonomies of content and process concepts in computer science (Zendler & Spannagel, 2008; Zendler, Spannagel, & Klaudt, 2009, Zendler, McClung, & Klaudt, 2012).

2.2 Central content concepts

In the first study (Zendler & Spannagel, 2008) 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)*¹. The content concepts were assessed by horizontal, vertical, time and sense criteria (cf. Schwill, 1994). The following 15 central content concepts were determined by cluster analysis: *algorithm, computer, data, problem, information, system, language, program, test, communication, software, process, model, computation und structure*. Definitions of the content concepts are provided in Appendix I.

2.3 Central process concepts

The second study (Zendler, Spannagel, & Klaudt, 2009) was undertaken on the background of *process as content* (Parker & 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*. Definitions of the process concepts are provided in Appendix I.

2.4 Data basis

The data matrix from the study by Zendler, McClung, and Klaudt (2012) is used as data basis (Figure 1). In that study, 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. Figure 1 shows the means of the content concepts in relation to process concepts as well as the means of the process concepts in relation to content concepts. The content and process concepts are each sorted in a descending direction with respect to the grand means (means of the content means, respectively, the means of the process means). The content concepts with the five highest rankings in the overall average are: problem, information, model, algorithm, and data. Those with the five lowest rankings are: program, test, communication, language, and computer. The process concepts with the five highest rankings in the overall average are: analyzing, classifying, problem solving, and problem solving and posing, categorizing as well as investigating. Those with the five lowest rankings are: transferring, communicating, presenting, collaborating and ordering.

2.5 Data analysis

Data is analyzed as follows: The 15×16 data matrix (see Figure 1) with the content concept means in

¹ Available at <http://www.acm.org/class/1998/> (last access on 12/05/2015).

relation to the process concepts serves as data basis.

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

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

In a first step, multi-dimensional scaling (MDS) using the ALSCAL algorithm (Takane, Young, & de Leeuw, 1977; Young, Takane, & Lewyckyj, 1978) is performed. For this, a distance matrix is developed for the content concepts, for which the process concepts serve as variables. They establish a 16-dimensional space in which the 15 content concepts are placed. The Euclidian distance is used as distance measure. The method of multi-dimensional scaling produces an MDS configuration of the content concepts. In a second step, a cluster analysis is performed for the MDS configuration of the content concepts. The authors applied Ward's method, a hierarchical procedure that calculates a series of cluster partitions (Everitt, Landau, & Leese, 2001). The Euclidian distance again served as distance measure. Multi-dimensional scaling and cluster analysis were calculated by SPSS 22.

3. Results

3.1 MDS configuration of the content concepts

Multi-dimensional scaling provides an MDS configuration of the content concepts with dimensions that can be interpreted semantically. As a method, multi-dimensional scaling is applied as structural analysis to graphically highlight similarities/dis-similarities among content concepts (see Borg & Groenen, 2005; Borg & Staufenbiel, 2007). Figure 1 shows the resulting MDS configuration of the content concepts in a 2-dimensional coordinate system. Placed in it are the content concepts with values for both dimensions. It should be noted that the content concepts *data* and *information* occupy similar positions in the coordinate system: both are characterized by high values in dimension #1 and high values in dimension #2. Also striking is the similar position of the two content concepts *communication* and *computer*: typical of both concepts is a relatively low value in dimension #1 and a relatively high value in dimension #2. Worth noting are also the comparable

positions of the three content concepts *test*, *software* and *language* and the two content concepts *system* and *computation*. Figure 2 illustrates that these content concepts could also be combined in groups.

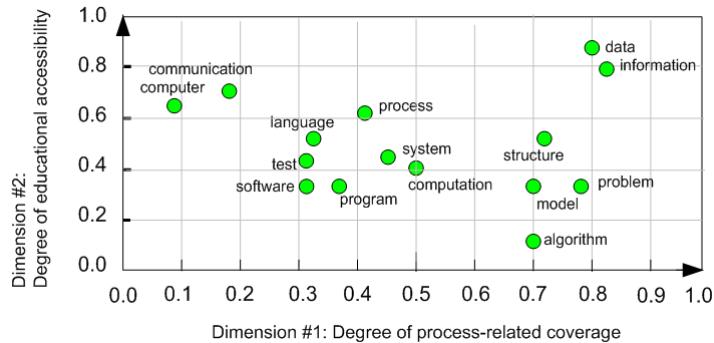


Figure 2. MDS configuration of content concepts

The MDS configuration was assessed by Young's S-stress formula 1 and the squared correlation of the distances with disparities R^2 (see Borg & Groenen, 2005). The following is obtained for the MDS configuration: S-stress $1 = .16$ and $R^2 = .87$. 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). For the interpretation of the dimensions of the MDS configuration, reference is made to the data set in Figure 1, for one, and to the relevant literature on computer science education, for another (Nievergelt, 1980, 1990; Schwill, 1994; Gal-Ezer & Harel, 1999; Tucker et al., 2003, 2006; Denning, 2003; GI, 2008; Zendler & Spannagel, 2008).

Dimension #1 - This dimension can be interpreted as *degree of process-related coverage*: Content concept with high values (> 2.50 threshold) for many process concepts have a high degree of process-related coverage (e.g., *information*, *model*, *problem*). Content concepts with high values for few process concepts have a low degree of process-related coverage (e.g., *communication*, *program*, *computer*). A content concept (see Figure 2) with very high degree of process-related coverage is *information*, for which the following 13 process concepts are of relevance: *analyzing*, *classifying*, *problem solving and posing*, *categorizing*, *investigating*, *finding relationships*, *generalizing*, *comparing*, *questioning*, *transferring*, *communicating*, *presenting* and *ordering*. A content concept with a very low degree of process-related coverage is *computer*, for which only five process concepts are of significance: *analyzing*, *classifying*, *categorizing*, *communicating* and *collaborating*.

Figure 3 depicts an example of the degree of process-related coverage for the content concepts *data* and *computer* with the aid of polar charts (Grinstein, Trutschl, & Cvek, 2001). The polar charts reveal that the larger the area spanned by the process concepts, the greater the degree of process-related coverage of a content concept.

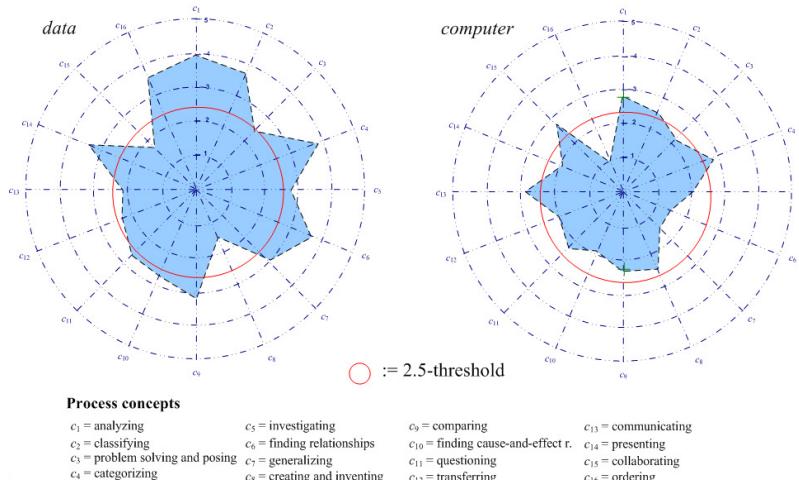


Figure 3. Polar charts for *data* and *computer*

Appendix II contains polar charts for each of the 15 central concept concepts.

Dimension #2 - This dimension can be interpreted as *educational accessibility*: Educational accessibility describes content 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 content concept with easy educational accessibility, for example, is *data*. This concept can be introduced in computer science education without reference to other content concepts of computer science. Besides, it can be introduced without having to take recourse to other resources and it can be taught on virtually any intellectual level. Examples of content concept with less easy educational accessibility include *algorithm*, *program* or *model*. Their introduction in computer science education depends on the prior introduction of other computer science content concepts (such as *data*, *computer*, *language*). Besides, some of them can only be dealt with if appropriate resources (software, hardware) are available.

Figure 4 depicts the content concepts upon which other content concepts are based. In the dependency graphs it is notable that the concept *data* has a central position and that four other concepts are directly based upon it (*information*, *program*, *process* and *algorithm*), whereas the concept *computer* is based upon a concept (*program*) directly, and that two other concepts are directly based upon it (*communication* and *computation*).

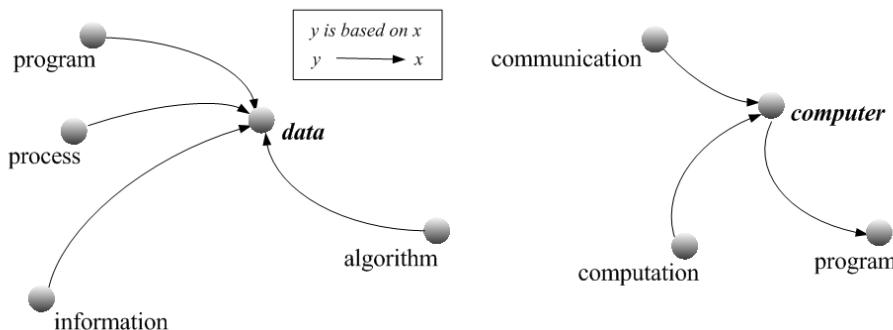


Figure 4. Dependency graphs for *data* and *computer*

Appendix II contains a dependency graph for the 15 content concepts.

3.2 Application of the two measures: competence areas determined

The MDS configuration showed that the content concepts can be represented with respect to two semantic dimensions, i.e., the degree of process-related coverage and the educational accessibility. Competence areas for computer science education are now identified by cluster analysis. This process uses the coordinates of the content concepts in the MDS configuration (see Figure 5): degree of process-related coverage and educational accessibility. Figure 4 shows the result of the cluster analysis: Totally four clusters are obtained for the 15 content 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.

Information technology - This competence area comprises the two content concepts *data* and *information* which are merged early due to their similar values in relation to the degree of process-related coverage and the educational accessibility. Typical of this competence area is easy educational accessibility and high degree of process-related coverage.

Model building - This competence area comprises the four content concepts *problem*, *model*, *structure*, and *algorithm*. The competence area has a high degree of process-related coverage but is not very easily accessible教育上. This is true, in particular, of *algorithm*. However, what is striking for this competence area is the early fusion of *problem* and *model*, whereas *structure* and *algorithm* cannot be assigned unless at some distance.

This implies certain heterogeneity of the concepts on the background of their degree of process-related coverage and their educational accessibility.

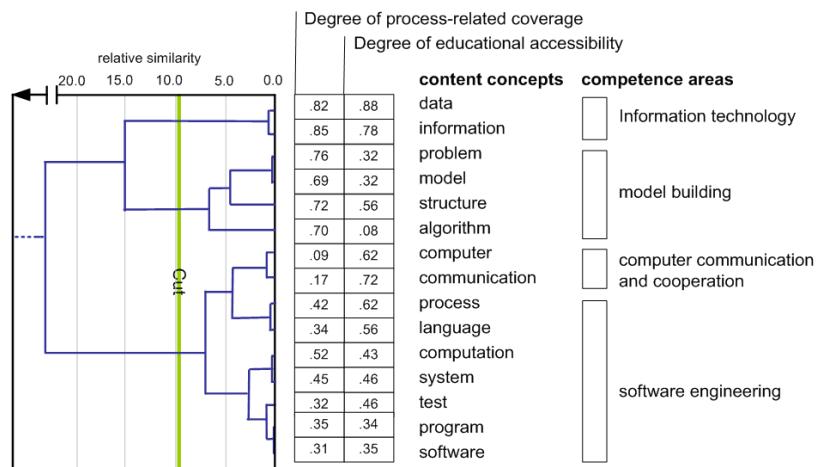


Figure 5. Assignment of content concepts to competence areas

Computer communication and cooperation - This competence area consists of the two content concepts *computer* and *communication*. Typical of this competence area is a low degree of process-related coverage and easy educational accessibility. It can be seen from the dendrogram that *computer* and *communication* are merged very early.

Software engineering - This competence area comprises the seven content concepts *process*, *language*, *computation*, *system*, *test*, *program* and *software*. It characterizes content concepts whose degree of process-related coverage and educational accessibility are in the mid-range.

4. Discussion

This study provided two important educational measures for the classification, comparison and usage of computer science content concepts: the degree of process-related coverage and the degree of educational accessibility. The two measures 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 applied in the context of the research-based training of computer science teachers (Abell, 2007; European Commission, 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 developing further measures to assess content and process concepts for computer science education. A starting point 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 hierachic relations.

5. References

- Abell, S. K. (2007). Research on science teacher knowledge. In S. K. Abell & N. G. Lederman, *Handbook of research on science education* (pp. 1105–1149). London, UK: Lawrence Erlbaum.
- Armoni, M., & Ginat, D. (2008). Reversing: A fundamental idea in computer science. *Computer Science Education*, 18(3), 213–230. <http://dx.doi.org/10.1080/08993400802332670>
- Bauer, F. L. (2009). *Die Lage der Informatik in der Bundesrepublik Deutschland*. Berlin, Germany: Springer.

- Bolhuis, S. (2003). Towards process-oriented teaching for self-directed lifelong learning: a multi-dimensional perspective. *Learning and Instruction* 12, 327–347. [http://dx.doi.org/10.1016/S0959-4752\(02\)00008-7](http://dx.doi.org/10.1016/S0959-4752(02)00008-7)
- 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.
- Bruner, J. S. (1960). *The process of education*. Cambridge: Harvard University 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, 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.
- Davis, E. A. (2004). Knowledge integration in science teaching: Analyzing teachers' knowledge development. *Research in Science Education*, 34, 21–53. <http://dx.doi.org/10.1023/B:RISE.0000021034.01508.b8>
- Denning, P. J. (2003). Great principles of computing. *Communications of the ACM*, 46(11), 15–20. <http://dx.doi.org/10.1145/948383.948400>
- Driel van, J. H., Beijaard, D., & Verloop, N. (2001). Professional development and reform in science education: The role of teachers' practical knowledge. *Journal of Research in Science Teaching*, 38(2), 137–158. [http://dx.doi.org/10.1002/1098-2736\(200102\)38:2<137::AID-TEA1001>3.0.CO;2-U](http://dx.doi.org/10.1002/1098-2736(200102)38:2<137::AID-TEA1001>3.0.CO;2-U)
- Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., Sanders, K., & Zander, C. (2006). Putting threshold concepts into context in computer science education. In *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 103–107). New York: ACM. <http://dx.doi.org/10.1145/1140123.1140154>
- Emmott, S. (2006). *Towards 2020 science*. Cambridge, UK: Microsoft.
- Ericson, B. (2008). *Ensuring exemplary teaching in an essential discipline. Addressing the crisis in computer science teacher certification*. New York, NY: ACM.
- 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 from http://ec.europa.eu/education/policies/2010/doc/principles_en.pdf
- Everitt, B. S., Landau, S., & Leese, M. (2001). *Cluster analysis*. London: Arnold.
- Fishman, B., & Davis, E. (2006). Teacher learning research and the learning sciences. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 535–550). New York, NY: Cambridge University Press.
- Franklin, B. M., & Johnson, C. C. (2008). What the schools teach. In F.M. Connelly, M.F. He, & J. Phillion (Eds.), *The SAGE handbook of curriculum and instruction* (pp. 460–477) Thousand Oaks, CA: Sage Publications. <http://dx.doi.org/10.4135/9781412976572.n23>
- Fuchs, C., & Hofkirchner, W. (2003). *Studienbuch Informatik und Gesellschaft*. Norderstedt: Books on Demand.
- Gal-Ezer, J., & Harel, D. (1999). Curriculum and course syllabi for a high-school CS program. *Computer Science Education*, 9(2), 114–147. <http://dx.doi.org/10.1076/csed.9.2.114.3807>
- 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.
- Grinstein, G., Trutschl, M., & Cvetk, U. (2001). *High-Dimensional Visualizations*. Retrieved from http://pdf.aminer.org/000/371/596/d_grand_tour_for_multidimensional_data_and_clusters.pdf
- 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 from http://germany.acm.org/upload/pdf/ACMandIE_report.pdf
- Johnson, D. G., & Miller, K. W. (2009). Ethical issues for computer scientists. In A. B. Tucker (Ed.), *Computer science handbook* (2nd ed.) (pp. 19–29). Boca Raton, FL: Chapman & Hall.
- Jones, S.P. (2011). *Computing at school. International comparisons*. Retrieved 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.
- 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>
- Nievergelt, J. (1980). Computer science education: An emerging consensus on basic concepts. In S. H. Lavington (Ed.), *Information Processing 80* (pp. 927–933). Amsterdam: North Holland.
- Nievergelt, J. (1990). Computer science for teachers: A quest for classics and how to present them. In D. H. Norrie & H. W. Six (Eds.), *Computer assisted Learning, Lecture Notes in Computer Science 438* (pp. 2–15). Berlin, Heidelberg, New York: Springer. <http://dx.doi.org/10.1007/BFb0020866>
- Parker, J. C., & Rubin, L. J. (1966). *Process as content. Curriculum design and the application of knowledge*. Chicago, IL: Rand McNally.
- Schwill, A. (1994). Fundamental ideas of computer science. *EATCS Buletin*, 53, 274–295.
- Shapiro, E., Harel, D., Bishop, C., & Muggleton, S. (2006). The fundamental role of computer science concepts in science. In S. Emmott (Ed.), *Towards 2020 science* (pp. 24–25). Redmond, CA: Microsoft.
- 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. B., & Wegner, P. (2009). The discipline and its impact. In A. B. Tucker (Ed.), *Computer science handbook* (2nd ed.) (pp. 3–18). Boca Raton, FL: Chapman & Hall.
- Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). *A model curriculum for K-12 computer science*. New York: ACM.
- Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2006). *A model curriculum for K-12 computer science* (2nd edition). New York: ACM.
- Wursthorn, B. (2005). Fundamental concepts of computer science in a Logo-environment. In G. Gregorczyk, A. Walat, Kranas, W., & Borowiecki, M. (Eds.), *Digital Tools for Lifelong Learning. Proceedings of the tenth European Logo Conference* (pp. 219–227). Warsaw: Centre for Informatics and Technology in Education.
- Young, F. W. Takane, Y., & Lewyckyj, R. (1978). Three notes on ALSCAL. *Psychometrika*, 43, 433–435. <http://dx.doi.org/10.1007/BF02293652>
- Zeichner, K. M. (1983). Alternative paradigms of teacher education. *Journal of Teacher Education*, 34(3), 3–9. <http://dx.doi.org/10.1177/002248718303400302>
- Zeidler, D. L. (2002). Dancing with maggots and saints: visions for subject matter knowledge, pedagogical knowledge, and pedagogical content knowledge in science teacher education reform. *Journal of Science Teacher Education*, 13(1), 27–42. <http://dx.doi.org/10.1023/A:1015129825891>
- 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., McClung, O. W., & Klaudt, D. (2012). Content and process concepts relevant to computer science education: A cross-cultural study. *International Journal of Research Studies in Computing*, 1(2), 27–47. <http://dx.doi.org/10.5861/ijrsc.2012.144>
- Zendler, A., Spannagel, C., & Klaudt, D. (2009). Process as content in computer science education: empirical determination of central processes. *Computer Science Education*, 18(4), 231–245.

APPENDIX I: Definitions of content and process concepts

Content concepts

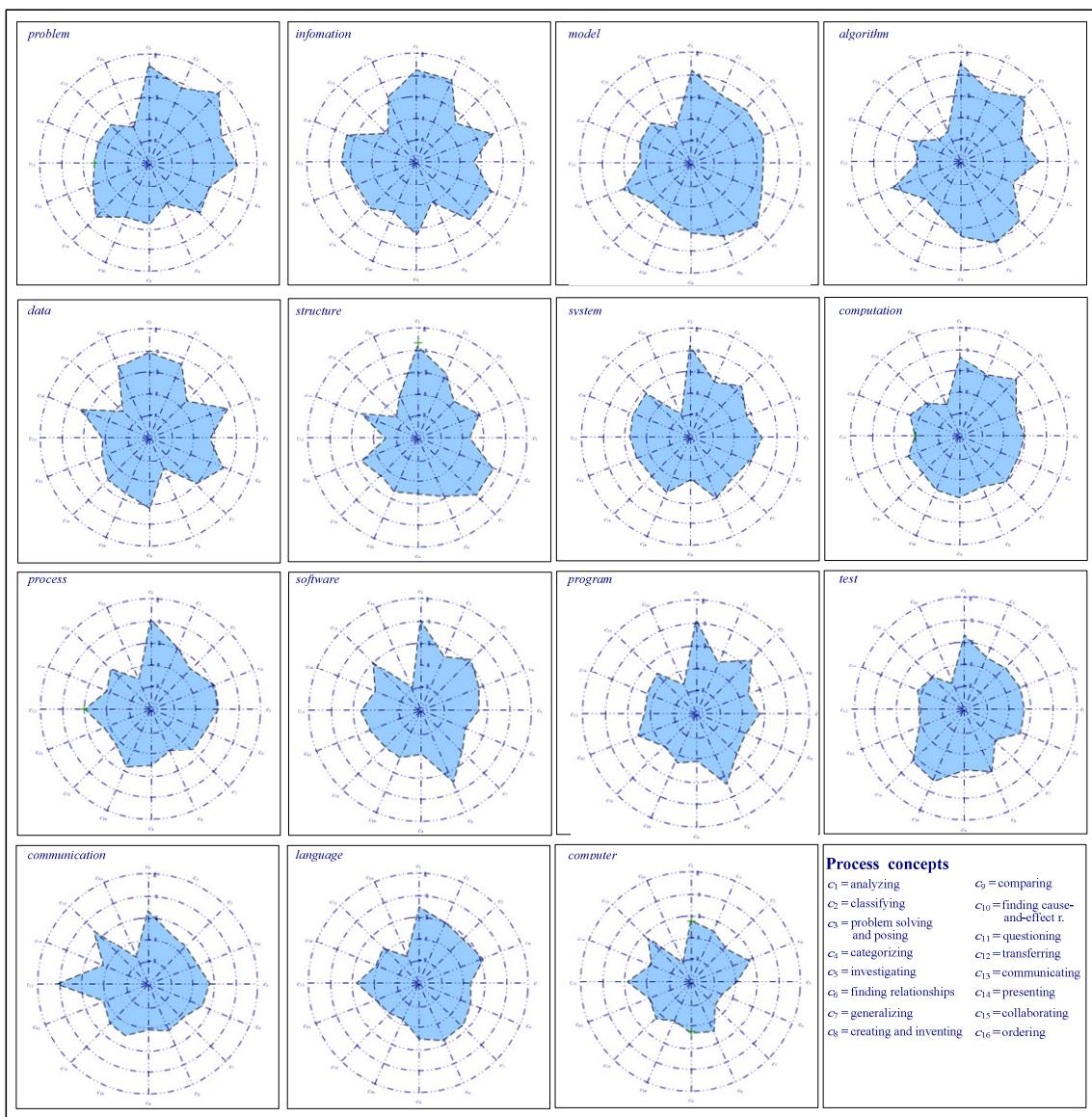
- Algorithm. An algorithm is a list of instructions to solve a class of problems.
- Communication. (Computer based) communication is understood as any form of human interaction by using two or several computers in a network.
- Computation. Computation is understood as a process that as a model is expressed 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

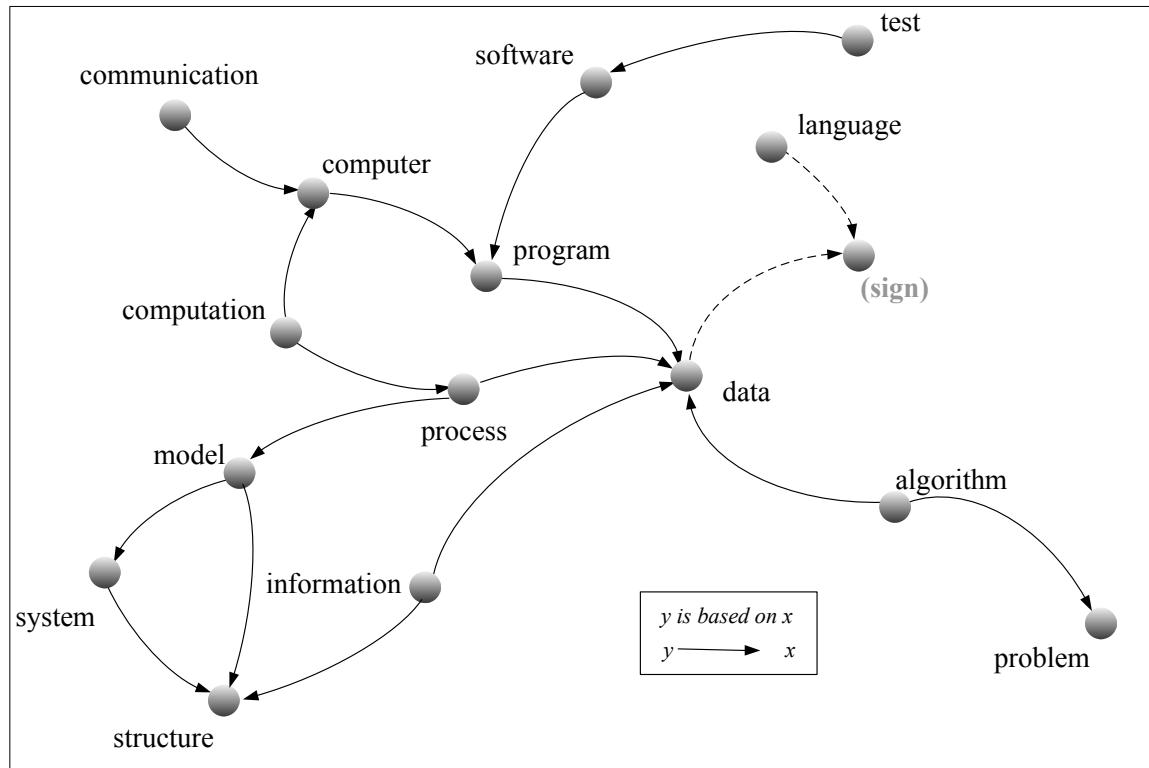
that are mutually productive by aligning personal goals with larger group goals, seeking consensus, resolving conflicts and disagreements with dignity and respect, valuing and protecting the common good, volunteering, and being of service to and feeling compassion for others.

- Communicating. Engaging in productive discussions, discourse, debate, and dialogue in their multiple forms (linguistic, visual, technological, kinesthetic, 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.

APPENDIX II:**Process-related coverage**

A-1. Process-related coverage - matrix of polar charts for the 15 content concepts

Educational accessibility



A-2. Educational accessibility - dependency graph of the 15 content concepts

