Empirical Foundation of Central Concepts for Computer Science Education

ANDREAS ZENDLER and CHRISTIAN SPANNAGEL

The design of computer science curricula should rely on central concepts of the discipline rather than on technical short-term developments. Several authors have proposed lists of basic concepts or fundamental ideas in the past. However, these catalogs were based on subjective decisions without any empirical support. This article describes the empirical determination of central concepts for computer science education. Experts of computer science rated 49 concepts regarding four criteria. The cluster analysis of the data revealed the following central concepts: *problem*, *data*, *computer*, *test*, *algorithm*, *process*, *system*, *information*, *language*, *communication*, *software*, *program*, *computation*, *structure*, and *model*.

Categories and Subject Descriptors: K.3.2 [**Computer and Education**]: Computer and Information Science Education—*Computer Science Education; Curriculum*

General Terms: Experimentation, Human Factors, Measurement, Standardization

Additional Key Words and Phrases: Central concepts, fundamental ideas

ACM Reference Format:
Zendler, A. and Spannagel, C. 2008. Empirical foundation of central concepts for computer science education. ACM J. Educ. Resour. Comput. 8, 2, Article 6 (June 2008), 15 pages. DOI = 10.1145. 1362787.1362790. http://doi.acm.org/10.1145.1362787.1362790.

1. INTRODUCTION

Curricula for computer science education in schools must rely on concepts of computer science that are important for general education. Since computer science is a permanently changing field, schools must promote long-lasting knowledge which is expected to be also relevant in the future. Thus, the fundamentals of computer science must be taken into account when curricula for computer science education are designed.

The proposal to design a curriculum in accordance with fundamental elements of the structure of a discipline has been made by Bruner [1960] and is still discussed today as a *science-oriented* approach [Schwill 1994; Eckerdal

Authors' addresses: A. Zendler and C. Spannagel, Institute for Mathematics and Computer Science, University of Education Ludwigsburg, Reuteallee 46, 71636 Ludwigsburg, Germany; email: zendler@ph-ludwigsburg.de.

et al. 2006]. According to Bruner [1960], constitutive elements of a discipline's structure are called *fundamental ideas*. Unfortunately, Bruner did not define the term *fundamental idea*; he also used the terms *fundamental idea*, *basic idea*, and *general principle* interchangeably. His remarks are more a sort of a slogan system which is somehow vague [Apple 1992].

Nevertheless, he describes some characteristics of fundamental ideas. They have "wide as well as powerful applicability" [Bruner 1960, p. 18], and they can be taught to children at any stage of development (cf., Schwill [1994]). Halmos [1981] compares basic ideas with chemical elements which can be combined and recombined to build up different forms of matter. The importance of fundamental ideas has already been emphasized by Whitehead in the late 1920s [1929].[1]

Many authors have published catalogs of fundamental ideas or basic concepts of computer science which are based on personal estimations [Nievergelt 1980, 1990; Schwill 1994; Denning 2003; Loidl et al. 2005; Wursthorn 2005]. For example, Schwill [1994] suggests that *algorithmization*, *structured dissection*, and *language* are fundamental ideas of computer science ("master ideas") which include numerous fundamental sub-ideas sorted in a hierarchical structure. Denning [2003] lists great principles and practices such as *computation*, *communication*, *performance*, and *modeling* and *validation*. Loidl et al. [2005] mention basic concepts such as *modeling*, *abstraction*, *states*, *algorithm*, *information*, and *language*. A serious point of criticism is that all lists rely on the appraisals of one or just a few experts. Thus, they are very subjective and are not more than proposals for discussion.

In addition, curricula such as the Computer Engineering Curriculum of IEEE/ACM [2004] or the Model Curriculum for K-12 Computer Science of the ACM [2003] rather provide an overview of discipline areas and topics than a list of central concepts of computer science.

What is needed is a *systematic* and *empirical* investigation where a list of central concepts is extracted from ratings of numerous experts. This article describes an empirical study in which experts of computer science have been systematically surveyed. The data have been analyzed by clustering techniques. This resulted in a catalog of central concepts for computer science education which is based on the opinion of many experts.

In the next section, the term *central concept* is clarified, including criteria which help to decide whether or not a concept belongs to the fundament of a discipline. In Section 3, the empirical study is described in detail and the results are discussed. This yields a list of 15 central concepts. Conclusions are drawn from the study in Section 4.

[1]Recently, the idea of *threshold concepts* has also been introduced to describe significant concepts in learning a specific subject matter [Meyer and Land 2005; Eckerdal et al. 2006; Boustedt et al. 2007]. Threshold concepts are *conceptual gateways* which transform the learner's perception of a subject. Rowbottom [2007] argues that there are methodological problems in identifying threshold concepts. Therefore the conception of threshold concepts has not been taken into consideration in our study.

2. CENTRAL CONCEPTS

A concept is a *unit of knowledge created by a unique combination of characteristics* [ISO 1087-1: 2000]. Concepts are distinguished from terms which designate concepts and from things and facts which are referenced by the concepts [Bunge 1967]. For instance, the sentence “The apple falls from the tree.” contains the word “apple” which designates the concept *apple*. The concept *apple* is an abstraction of all apples of the past, present, and future. It incorporates the attributes of these fruits like form and color.

Idea is much more difficult to define than *concept*. Schwill [2004] discusses differences between idea and concept on a philosophical background. He states that the term “concept” in conjunction with a fundamentalist orientation to the structure of a discipline might have led to aberrations in mathematics education in the past (set theory in primary schools in the 1960s). For him, ideas “are idealized imaginations which objectives are attached to that may not be experienced. However, they guide the human impulse to research and instruct the mind to extend its knowledge towards these objectives possibly without ever reaching them.” [Schwill 2004, p. 154] Thus, *idea* is much vaguer than *concept*.

To be concrete, we use the term *concept* according to the ISO standard. Particularly, we speak of *central concepts* to emphasize that they should be focussed when designing courses and curricula. We also prefer the term *central concept* because it has no fundamentalist connotation.

What are the characteristics of central concepts? We propose to use Schwill's four criteria[2] for fundamental ideas of computer science to determine whether or not any given concept is a central concept. The criteria are (according to Schwill [1994]):

—*Horizontal criterion*: The concept is applicable or observable in multiple ways in different areas of the domain.
—*Vertical criterion*: The concept may be demonstrated and taught at every intellectual level.
—*Criterion of time*: The concept can be clearly observed in the historical development of the domain and will be relevant in the longer term.
—*Criterion of sense*: The concept is related to everyday language and/or thinking.

The horizontal criterion and the vertical criterion are based on Bruner's characteristics of fundamental ideas [Bruner 1960]. The vertical criterion directly leads to the concept of the *spiral curriculum* in which topics are revisited several times throughout a curriculum or a course [Harden and Stamper 1999]. The criterion of time is especially important for computer science where long-term concepts have to be distinguished from short-term developments. “*How*

[2]Two further criteria can be found in literature in addition to Schwill's four original criteria: the *goal criterion* [Schwill 2004] and the *criterion of representation* [Hartmann et al. 2006]. We used the four original criteria because they are part of all criteria catalogs in the context of computer science education.

do we recognize ideas of long-lasting value among the crowd of fads? The "test of time" is the most obvious selector. Other things being equal, ideas that have impressed our predecessors are more likely to continue to impress our successors than our latest discoveries will" [Nievergelt 1990, p. 5] (emphasis in the original). The criterion of sense has originally been proposed by Schreiber [1983] in the domain of mathematics: To understand the sense of a concept, students should be able to relate the concept to everyday experiences.

The four criteria are used to develop a questionnaire which allows for the empirical identification of central concepts. The empirical study is described in the next section.

3. EMPIRICAL STUDY

3.1 Method

3.1.1 *Sample.* Our questionnaire was sent to 98 experts of computer science, professors working at 14 universities in Germany. Only universities with the best reputation at research had been selected (according to the CHE ranking 2006[3]; e.g., the Technical University of Munich, the University of Karlsruhe, and RWTH Aachen University). Thirty-seven valid questionnaires have been sent back (approximately 37.8 percent).

3.1.2 *Questionnaire.* The questionnaire consisted of a short introduction, followed by four structurally equivalent parts. Each part was related to one criterion (*horizontal criterion*, *vertical criterion*, *criterion of time*, and *criterion of sense*) and contained 49 concepts of computer science. In those sections, the subjects had to rate to what extent the concepts meet the particular criterion. For instance, in the section related to the horizontal criterion the subjects had to rate the following statement on a scale from 0 (*strongly disagree*) to 5 (*strongly agree*) for each concept: "The concept is applicable or observable in multiple ways in different areas of the domain."

There are two essential documents which can be used as basis for the creation of the questionnaire: The *ACM Computing Classification System (1998 Version)*[4] and the *Computer Engineering Curriculum* [IEEE/ACM 2004] which contains a comprehensive body of knowledge. Concepts contained in the *Computer Engineering Curriculum* have already been filtered with regard to educational demands. However, the questionnaire must represent the discipline and not existing curricula. Therefore, concepts were taken from the *ACM Computing Classification System* in order to obtain a pure representation of the discipline.

Only those concepts were used which are mentioned more than 10 times in the *ACM Computing Classification System*. Phrases and inflected forms of words were taken into consideration. For instance, the phrase *distributed objects* was ranked among the concept *distribution* and the concept *object*. The 49

[3]Retrieved September 11, 2006, from http://www.che.de
[4]Retrieved September 11, 2006, from http://www.acm.org/class/1998/

Table I. The 49 Concepts of the Questionnaire and the Number of their Occurrences in the ACM Computing Classification System (1998 Version)

No.	Concept	Occurrences	No.	Concept	Occurrences
1	system	75	26	simulation	17
2	language	57	27	generation	17
3	program	54	28	representation	16
4	method/methodology	48	29	code	15
5	design	47	30	reliability	15
6	data	43	31	parallelism	15
7	computer	39	32	equation	15
8	analysis*	37	33	device	15
9	model	32	34	graphics	15
10	software	30	35	application	14
11	computation	29	36	interface	14
12	theory	28	37	error	14
13	test	28	38	verification	14
14	network	27	39	problem	13
15	control	26	40	performance	13
16	architecture	25	41	technique	12
17	structure	24	42	management***	12
18	information	24	43	standard	12
19	logic	22	44	hardware	12
20	algorithm	22	45	automation	12
21	processor	21	46	communication	11
22	distribution**	21	47	statistics	11
23	memory	21	48	approximation	11
24	database	19	49	process	11
25	processing	19			

*analysis in a general sense (not the subdiscipline of mathematics)
**distribution in terms of distributed applications
***management in terms of product or process management

concepts are listed together with the number of their occurrences in Table I.[5] In the questionnaire they were presented in random order.

3.1.3 *Data Analysis.* In a first step, concepts which are similar regarding the four criteria are determined by clustering techniques. This leads to clusters of concepts which have similar rankings. To assure that the clusters form a proper discrimination of the concepts, multivariate comparisons of means are conducted.

3.2 Results

First, the results of the descriptive data analysis are reported, followed by the results of the cluster analysis. In the last part of this section, the results regarding the stability and validity of the cluster solution are explained. SPSS 12.0 has been mainly used for the analysis of the data. The procedure to compute the stability has been implemented in R[6].

[5]A similar quantitative approach was undertaken by Armstrong [2006] in the field of object-oriented development.
[6]R is a language for statistical computing and can be retrieved from http://www.r-project.org/ (last retrieved July 14, 2007).

horizontal criterion	vertical criterion	criterion of time	criterion of sense	*total score*	concepts
4.24	4.08	4.62	3.54	4.12	algorithm
3.57	4.05	4.46	4.14	4.06	computer
3.84	3.89	4.41	3.95	4.02	data
3.70	3.86	4.30	4.08	3.99	problem
3.86	3.46	4.38	4.19	3.97	information
3.97	3.32	4.35	3.78	3.86	system
3.76	3.19	4.35	4.03	3.83	language
3.65	3.54	4.32	3.59	3.78	program
3.49	3.78	3.73	4.05	3.76	test
3.46	3.35	4.16	3.89	3.72	communication
3.70	3.62	4.35	3.22	3.72	software
3.69	3.11	4.19	3.65	3.66	process
4.05	3.00	4.19	3.14	3.60	model
3.51	3.24	4.30	3.27	3.58	computation
2.84	3.16	4.14	4.05	3.55	network
3.00	3.54	3.89	3.78	3.55	error
2.78	3.59	4.30	3.43	3.52	hardware
3.70	3.32	3.76	3.24	3.51	structure
3.46	2.62	4.11	3.27	3.37	architecture
3.11	3.00	3.84	3.24	3.30	application
3.57	3.03	3.84	2.68	3.28	representation
3.65	2.49	3.95	2.89	3.25	method/methodology
3.24	3.00	3.81	2.86	3.23	interface
3.03	2.49	3.84	3.51	3.22	reliability
3.11	2.70	3.76	3.24	3.20	performance

horizontal criterion	vertical criterion	criterion of time	criterion of sense	*total score*	concepts
2.76	2.86	3.92	3.16	3.18	distribution
3.81	2.43	4.08	2.41	3.18	theory
2.76	3.03	3.84	3.05	3.17	parallelism
3.30	2.68	3.97	2.70	3.16	processing
2.38	3.08	3.92	3.08	3.12	memory
3.35	2.32	3.68	3.03	3.10	design
2.70	2.86	4.08	2.54	3.05	processor
2.57	2.70	3.92	2.97	3.04	database
3.14	2.49	3.95	2.43	3.00	logic
3.00	2.81	3.19	2.76	2.94	technique
2.51	2.41	3.59	3.14	2.91	automation
3.35	2.27	3.38	2.51	2.88	analysis
2.97	2.42	3.25	2.81	2.86	control
2.78	2.78	3.92	1.97	2.86	code
2.19	2.62	3.51	3.08	2.85	graphics
3.14	2.22	3.84	2.00	2.80	verification
2.30	2.54	3.22	3.11	2.79	standard
2.68	2.03	3.54	2.47	2.68	simulation
1.95	2.32	3.32	2.51	2.53	device
1.92	1.70	3.11	3.11	2.46	management
2.59	1.65	3.16	1.76	2.29	generation
1.92	2.16	2.54	2.11	2.18	approximation
1.95	2.14	2.95	1.65	2.17	equation
1.81	1.65	2.32	2.59	2.09	statistics

Fig. 1. Means of the concepts ($N = 37$).

3.2.1 *Descriptive Data Analysis.* Figure 1 shows the means of the concepts regarding the horizontal criterion, the vertical criterion, the criterion of time, and the criterion of sense. An overall score was calculated as the mean of the four criterion values. This value was used to sort the concepts in descending order.

Concepts with the five highest overall values are *algorithm*, *computer*, *data*, *problem*, and *information*. Concepts with the five lowest overall values are *management*, *generation*, *approximation*, *equation*, and *statistics*.

3.2.2 *Cluster Analysis.* The cluster analysis is based on the means of the concepts regarding the four criteria. The cluster analysis according to Ward [1963] has been performed. This is a hierarchical procedure which computes a sequence of partitions [Everitt et al. 2001]. As a distance measure, the squared Euclidean distance has been used.

The cluster analysis revealed eight clusters. The termination of the computation (visualized as "cut" in the following figures) was performed in consideration of the C-Index according to Hubert and Levin [1976]. In the following sections, the eight clusters are described. For purposes of clarity, clusters are combined in "winner" clusters (short: W clusters), "intermediate" clusters (short: I clusters), and "loser" clusters (short: L clusters). Generally speaking,

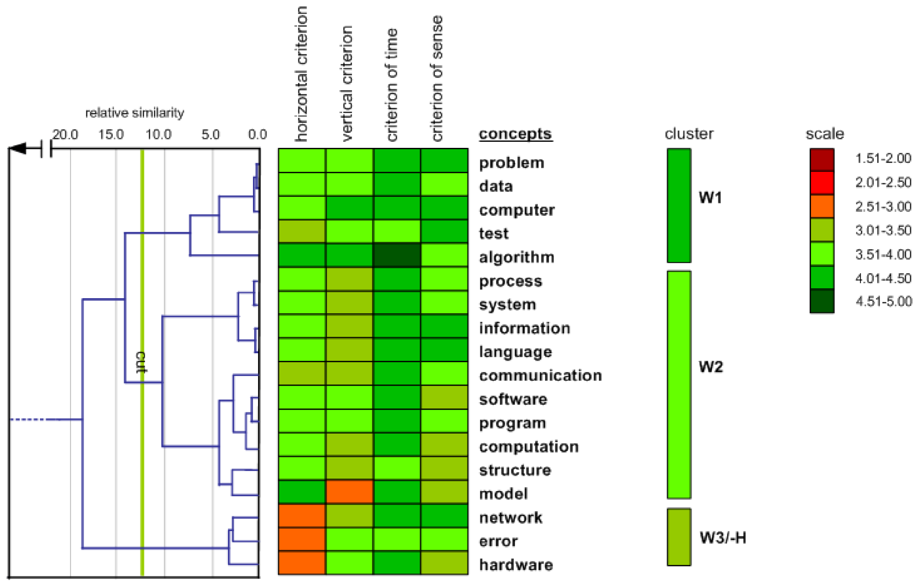

Fig. 2. The "winner" clusters.

the W clusters contain concepts that are important for computer science education regarding the four criteria, the L clusters contain concepts that are unimportant for computer science education, and the I clusters contain concepts that could be interesting in particular cases.

W clusters. Figure 2 shows the W clusters which are the three clusters W1, W2, and W3/-H. The heat map in Figure 2 visualizes the numerical values of Figure 1 regarding the four criteria. In addition, the partitions produced at subsequent steps are shown in the dendrogram. The dendrogram shows the sequence of fusions according to the relative similarity of the concepts with regard to the four criteria.

W1 cluster and W2 cluster. These clusters are characterized by concepts that have the highest values regarding the horizontal criterion, the vertical criterion, the criterion of time, and the criterion of sense, compared to the concepts of the other clusters. The two subclusters W1 and W2 consist of 15 concepts overall. The W1 cluster contains the concepts *algorithm*, *computer*, *data*, *problem*, and *test* [centroid of the W1 cluster = (3.77, 3.93, 4.30, 3.95)]. Concepts in the W2 cluster clearly have lower means regarding the vertical criterion or the criterion of sense than the concepts in the W1 cluster [centroid of the W2 cluster = (3.74, 3.32, 4.24, 3.60)]. Concepts in the W2 cluster are *information*, *system*, *language*, *program*, *communication*, *software*, *process*, *model*, *computation*, and *structure*. The dendrogram shows that *problem*, *data*, and *computer* are grouped first due to their similar values regarding all criteria. *Process* and *system* as well as *information* and *language* are grouped quite soon in the W2 cluster. In addition, *software*, *program*, *computation*, and

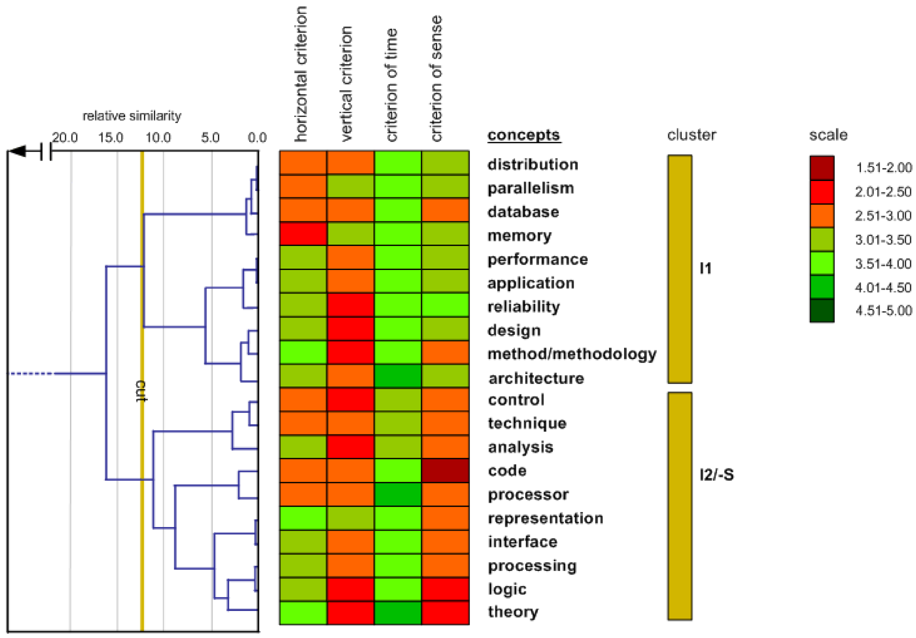

Fig. 3. The "intermediate" clusters.

communication are fused soon just like *structure* and *model*. The concept *test* is included quite late in the W1 cluster, just as *algorithm* which has the highest values regarding all four criteria.

W3/-H cluster. This cluster is designated by concepts having high values regarding the vertical criterion, the criterion of time, and the criterion of sense, but low values in the horizontal criterion [centroid of the W3/-H cluster = (2.87, 3.43, 4.11, 3.75)]. The cluster consists of three concepts: *network*, *error*, and *hardware*. The data reveals that the concepts have a similar profile like the concepts in the W1 cluster and the W2 cluster (except the horizontal criterion), but with lower values.

I clusters. Figure 3 shows the cluster solution of the I cluster which consists of two subclusters I1 and I2/-S. The I cluster is the biggest cluster and includes 20 concepts.

I1 cluster. This cluster is characterized by concepts that have medium values regarding the criterion of sense, relatively low values regarding the vertical criterion and relatively high values regarding the criterion of time [centroid of the I1 cluster = (3.02, 2.73, 3.88, 3.14)]. The I1 cluster consists of the following concepts: *distribution*, *parallelism*, *database*, *memory*, *performance*, *application*, *reliability*, *design*, *method/methodology*, and *architecture*. *Distribution*, *parallelism*, *database*, and *memory* are fused early. Similarly, the concepts *performance*, *application*, *reliability*, *design*, *method/methodology*, and *architecture* are grouped at an early stage.

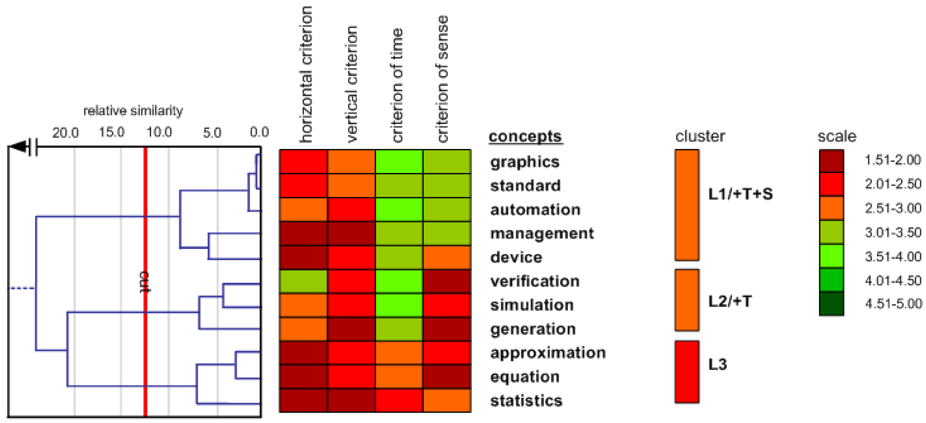

Fig. 4. The "loser" clusters.

I2/-S cluster. This cluster is specified by concepts that have similar values regarding the horizontal criterion, the vertical criterion and the criterion of time as the concepts in the I1 cluster. But compared to the concepts in the I1 cluster which have high values regarding the criterion of sense, the concepts in the I2/-S cluster have low values regarding this criterion [centroid of the I2/-S cluster = (3.19, 2.68, 3.75, 2.57)]. The I2/-S cluster consists of the concepts *control*, *technique*, *analysis*, *code*, *processor*, *representation*, *interface*, *processing*, *logic*, and *theory*. The concepts *representation* and *interface* are grouped early, as are the concepts *processing* and *logic*.

L clusters. Figure 4 shows the cluster solution of the L cluster which consists of three subclusters L1/+T+S, L2/+T, and L3. The L cluster covers 11 concepts. The dendrogram and the heat map visualize the structure of the cluster.

L1/+T+S cluster. This cluster is designated by concepts that have relatively low values regarding the horizontal and the vertical criterion but relatively high values regarding the criterion of time and the criterion of sense. The L1/+T+S cluster contains the concepts *graphics*, *standard*, *automation*, *management*, and *device* [centroid of the L1/+T+S cluster = (2.17, 2.32, 3.35, 2.99)]. The concepts *graphics*, *standard*, and *automation* are very similar with regard to their ratings and thus are grouped together early. Contrarily, *management* and *device* are added at a later stage as a result of their dissimilarity.

L2/+T cluster. This cluster is defined by concepts that have low values regarding the vertical criterion and the criterion of sense. But compared to the concepts in the L3 cluster, they have relatively high values regarding the criterion of time [centroid of the L2/+T cluster = (2.80, 1.97, 3.51, 2.08)]. The concepts of L2/+T cluster are *verification*, *simulation*, and *generation*.

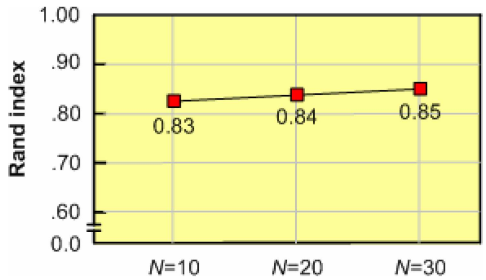

Fig. 5. Rand index depending on N.

L3 cluster. This cluster is characterized by concepts that have the lowest values regarding all four criteria [centroid of the L3 cluster = (1.89, 1.98, 2.60, 2.12)]. Concepts in this cluster are *approximation*, *equation*, and *statistics*.

3.2.3 *Stability and Validity of the Cluster Solutions.* To test the results of the cluster analysis, the stability and validity of the results have been checked.

Stability. The stability of the results has been tested with the Rand index RI [Rand 1971; Hubert and Arabie 1985]. Three additional cluster analyses have been performed with N = 10, N = 20, and N = 30. The results of these analyses have been compared with the results of the cluster analysis with N = 37. Figure 5 shows the Rand indices depending on the number of data cases involved. The comparison of the results with N = 10 already revealed a high similarity. The similarity grows for N = 20 and N = 30.

Validity. The validity has been tested by conducting a one-factorial four-variate analysis of variance (MANOVA) with subsequent post-hoc tests. The MANOVA has been performed to check if the eight clusters differ significantly regarding to all four criteria. The post-hoc tests were conducted to determine the clusters which have significant differences.

As a test statistics for the MANOVA, Wilk's Λ has been used and transformed to F statistics. The empirical F statistics were 11.85, which is larger than the critical value $F_{(28,138)}$= 1.80 (α = .01). That means that the eight clusters differ significantly.

Similarly, Wilk's Λ has been used as a test statistics for the post-hoc tests, and again F statistics have been computed. These values can be seen in Figure 6. In each case the degrees of freedom for the nominator and the denominator are specified. In the right part of Figure 6, the critical values for α = .01 and α = .05 are listed. It can be concluded that 20 of 28 post-hoc tests have been significant at α = .01 (**), and 6 have been significant at α = .05 (*). Two comparisons were not significant: the comparison of L2/+T and L3 and the comparison of L2/+T and W3/-H. To summarize, the statistical tests show that the eight clusters differ clearly.

	W2	W3/-H	I1	I2/-S	L1/+T+S	L2/+T	L3
W1	11.20** (4;10)	14.12* (4; 3)	75.47** (4; 10)	41.91** (4; 10)	29.32** (4; 5)	59.39** (4; 3)	193.36** (4; 3)
W2		10.76** (4; 8)	30.54** (4; 15)	22.74** (4; 15)	41.91** (4; 10)	31.99** (4; 8)	92.17** (4; 8)
W3/-H			12.49** (4; 8)	16.36** (4; 8)	7.97* (4; 3)	8.38 (4; 1)	281.10* (4; 1)
I1				11.53** (4; 15)	10.16** (4; 10)	13.46** (4; 8)	79.88** (4; 8)
I2/-S					12.61** (4; 10)	3.99* (4; 8)	16.11** (4; 8)
L1/+T+S						8.17* (4; 3)	15.97* (4; 3)
L2/+T							5.06 (4; 1)

df	$F_{crit, .99}$	$F_{crit, .95}$
(4; 1)	5624.26	224.58
(4; 3)	28.71	9.12
(4; 5)	11.39	5.19
(4; 8)	7.01	3.84
(4; 10)	5.99	3.48
(4; 15)	4.89	3.06

Fig. 6. Calculated and critical F values of the post-hoc tests.

3.3 Discussion

Almost all concepts yielded high values regarding the criterion of time (exceptions: *approximation*, *equation*, and *statistics*). This indicates that with the quantitative analysis of the *ACM Computing Classification System* only concepts have been identified which are relevant for a longer period of time.

The W1 cluster includes important central concepts: *algorithm*, *computer*, *data*, *problem*, and *test*. The most important concept is *algorithm*. This concept reached the first rank in three criteria: the horizontal criterion, the vertical criterion, and the criterion of time. Regarding the criterion of time, it has the highest value overall (4.62). Because of their high values regarding all criteria, the following conclusions can be drawn for the concepts in the W1 cluster: They are relevant in all areas of computer science (horizontal criterion); they can be taught on every intellectual level (vertical criterion); they are long-lasting (criterion of time); and they are related to everyday thinking (criterion of sense). Concepts in the W2 cluster apparently have lower values regarding the vertical criterion than the concepts in the W1 cluster. That means that it is more difficult to teach them on every class level.

Many concepts in the W clusters are mentioned in literature on fundamental ideas or basic concepts of computer science, especially *algorithm* and *language* [Schwill 1994], *computation* and *communication* [Denning 2003], and *data* and *information* [Loidl et al. 2005]. Moreover, the concept *model* seems not to play the outstanding role as it has often been proposed [Thomas 2002; Hubwieser and Broy 1999; Wursthorn 2005]. The concepts *network*, *error*, and *hardware*, which belong to the W3/-H cluster, are not relevant to all subdisciplines of computer science (low values regarding the horizontal criterion), but they seem to be important in those areas where they can be found (high values regarding the other criteria).

It seems that concepts in the I clusters cannot be taught at every intellectual level, because they have low values regarding the vertical criterion. Exceptions are *parallelism*, *memory*, and *representation*. In addition, concepts in the I2/-S cluster have low values regarding the criterion of sense. This indicates that these concepts don't play a role in everyday life. They seem to be of technical kind (exceptions: *representation*, *theory*) so that the I2/-S cluster could also be named the "cluster of technical concepts." Perhaps those concepts are adequate for computer science lessons in schools which are technically oriented.

Concepts in the L clusters have low values regarding the horizontal and vertical criterion, *management* and *statistics* in particular. It can be concluded that those concepts are only relevant to advanced courses in computer science. Concepts in the L1/+T+S cluster are related to everyday language or thinking. Thus, it is important to discuss those concepts at a certain point in time, even though they are not relevant to many areas of computer science. This especially applies to the concepts *graphics*, *standard*, *automation*, and *management*.

4. CONCLUSION AND FUTURE WORK

This study yielded an empirical catalog of central concepts for computer science education. Based on the empirical findings of this study it can be concluded that the following concepts are the central concepts for computer science education: *problem*, *data*, *computer*, *test*, *algorithm*, *process*, *system*, *information*, *language*, *communication*, *software*, *program*, *computation*, *structure*, and *model*. Especially the concept *algorithm* seems to play the most important role in computer science education. Interestingly, the concept *model* has a low rank among the central concepts. Although it is part of the list of central concepts, it seems not be as important as it is often suggested.

The existence of clusters shows that there are concepts of equal importance for the design of computer science curricula. From the results it can be concluded that especially the concepts of the W1 cluster should be focused first when planning computer science courses. Of course, instruction regarding central concepts must be adapted to learner characteristics (e.g., prior knowledge) and teaching objectives. For example, at the university of education Ludwigsburg this has been done with respect to the concept *system* [Brosowski et al. 2006; Mengesdorf et al. 2007; and Höllwarth et al. 2007]. In subsequent work the central concepts have to be specified in more detail for computer science education. For example, this may be done by preparing a semantic net that relates central concepts to sub-concepts and describes their properties. Important contributions in the context of building conceptual and didactical systems are Brinda and Schubert [2001]; Mead et al. [2006]; Meyer [2006]; Cassel et al. [2007], and Pedroni et al. [2007]. It would also be interesting to survey instructors of computer science and practitioners in industry. This would perhaps yield different results with the need for modified or even new interpretations. Finally, we are interested in doing the survey as a multinational study to investigate national differences.

Moreover, *processes* and *methods* (*process as content*; cf., Parker and Rubin [1966]; Costa and Liebmann [1997]) which are specific to computer science (e.g., problem solving, analyzing, classifying) should be associated with the central concepts of computer science. This leads to a curriculum where concepts and processes are integrated. An example of such a curriculum is the National Council of Teachers of Mathematics (NCTM) standards for mathematics [NCTM 2000] that include not only mathematical contents (e.g., *number and operations*, *measurement*, or *data analysis and probability*) but also mathematical processes and methods (e.g., *problem solving* or *reasoning and proof*). Central processes for computer science education can also be empirically determined by surveying experts with a questionnaire similar to the one used in this study. The empirical determination of central processes for computer science education is the next step toward a computer science curriculum which is justified by empirical evidence.

ACKNOWLEDGMENT

We would like to thank Thomas Reuter for his kindly support with the data collection and Professor Laura Martignon for her helpful suggestions.

REFERENCES

ACM. 2003. A model curriculum for K-12 computer science. Final report of the ACM K-12 task force curriculum committee. Available from http://www.acm.org/education/curricula.html.

APPLE, M. W. 1992. Do the standards go far enough? Power, policy, and practice in mathematics education. *J. Res. Math. Educ. 23*, 5, 412–431.

ARMSTRONG, D. 2006. The quarks of object-oriented development. *Comm. ACM 49*, 2, 123–128.

BOUSTEDT, J., ECKERDAL, A., MCCARTNEY, R., MOSTRÖM, J. E., RATCLIFFE, M., SANDERS, K., AND ZANDER, C. 2007. Threshold concepts in computer science: Do they exist and are they useful? In *Proceedings of the 38th Technical Symposium on Computer Science Education (SIGCSE'07)*. ACM, New York, 504–508.

BRINDA, T. AND SCHUBERT, S. E. 2001. Didactic system for object-oriented modeling. In *Proceedings on the 7th World Conference on Computers in Education (WCCE'01)*. Copenhagen. IFIP, 473–482.

BROSOWSKI, S., GIERZ, G. R., AND SPANNAGEL C. 2006. Der Systembegriff—Beispielhafte Umsetzung des zentralen Informatikkonzepts System im Schulunterricht. *Notes on Educational Informatics—Section B: Classroom Experiences, 1*, 1–10.

BRUNER, J. S. 1960. *The Process of Education*. Harvard University Press, Cambridge, MA.

BUNGE, M. 1967. *Scientific Research I. The Search for System*. Springer, Berlin, Germany.

CASSEL, L., SLOAN, R., DAVIES, G., TOPI, H., AND MCGETTRICK, A. 2007. The computing ontology project—the computing education application. *SIGCSE Bull. 39*, 1, 519–520.

COSTA, A. L. AND LIEBMANN, R. M. 1997. *Envisioning Process as Content. Toward a Renaissance Curriculum*. Corwin Press, Thousand Oaks, CA.

DENNING, P. J. 2003. Great principles of computing. *Comm. ACM 46*, 11, 15–20.

ECKERDAL, A., MCCARTNEY, R., MOSTRÖM, J. E., RATCLIFFE, M., SANDERS, K., AND ZANDER, C. 2006. Putting threshold concepts into context in computer science education. In *Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education (SIGCSE'06)*. ACM, New York, 103–107.

EVERITT, B. S., LANDAU, S., AND LEESE, M. 2001. *Cluster Analysis*. Arnold, London, UK.

HALMOS, P. R. 1981. Does mathematics have elements? *The Math. Intell. 3*, 147–153.

HARDEN, R. M. AND STAMPER, N. 1999. What is a spiral curriculum? *Medical Teacher 21*, 2, 141–143.

HARTMANN, W., NÄF, M., AND REICHERT, R. 2006. *Informatikunterricht planen und durchführen*. Springer, Berlin, Germany.

HÖLLWARTH, K., KAMADJONG, C. T., AND KLAUDT, D. 2007. Das Betriebssystem: eine Einführung—Beispielhafte Umsetzung des zentralen Informatikkonzepts System im Schulunterricht. *Notes on Educational Informatics—Section B: Classroom Experiences, 2*, 23–35.

HUBERT, L. J. AND LEVIN, J. R. 1976. A general statistical framework for assessing categorical clustering in free recall. *Psych. Bull. 83*, 1072–1080.

HUBERT, L. J. AND ARABIE, P. 1985. Comparing partitions. *J. Classif. 2*, 193–218.

HUBWIESER, P. AND BROY, M. 1999. Educating surfers or craftsmen: Introducing an ICT curriculum for the 21st century. In *Communications and Networking in Education: Learning in a Networked Society. IFIP WG 3.1 and 3.5 Open Conference*. Aulanko-Hämeenlinna, Finland, T. Downes and D. Watson, Eds.

IEEE/ACM. 2004. Computer Engineering 2004. Curriculum guidelines for undergraduate degree programs in computer engineering. Available from http://www.acm.org/education/curricula.html

ISO 1087-1. 2000. *Terminology Work—Vocabulary—Part 1: Theory and Application*. International Organization for Standardization, Geneva, Switzerland.

LOIDL, S., MÜHLBACHER, J., AND SCHAUER, H. 2005. Preparatory knowledge: Propaedeutic in informatics. In *From Computer Literacy to Informatics Fundamentals*, R. T. Mittermeir, Ed. Springer, Berlin, Germany, 105–115.

MEAD, J., GRAY, S., HAMER, J., JAMES, R., SORVA, J., CLAIR, C. ST., AND THOMAS, L. 2006. A cognitive approach to identifying measurable milestones for programming skill acquisition. *ACM SIGCSE Bull. 38*, 4, 182–194.

MENGESDORF, F., TÖNSMANN, L., AND ZENDLER, A. 2007. Computernetzwerk—Beispielhafte Umsetzung des zentralen Informatikkonzepts System im Schulunterricht. *Notes on Educational Informatics—Section B: Classroom Experiences, 2*, 13–22.

MEYER, B. 2006. Testable, reusable units of cognition. *IEEE Computer 39*, 4, 20–24.

MEYER, J. H. F. AND LAND, R. 2005. Threshold concepts and troublesome knowledge (2): Epistemological considerations and a conceptual framework for teaching and learning. *Higher Education, 49*, 373–388.

NATIONAL COUNCIL OF TEACHERS OF MATHEMATICS. 2000. *Principles and Standards for School Mathematics*. NCTM, Reston, VA.

NIEVERGELT, J. 1980. Computer science education: An emerging consensus on basic concepts. In *Information Processing 80*, S. H. Lavington, Ed. North Holland, Amsterdam, 927–933.

NIEVERGELT, J. 1990. Computer Science for Teachers: A quest for classics and how to present them. Lecture Notes in Computer Science, vol. 438, D. H. Norrie and H. W. Six, Eds. Springer, Berlin, Germany, 2–15.

PARKER, J. C. AND RUBIN, L. J. 1966. *Process as Content. Curriculum Design and the Application of Knowledge*. Rand McNally, Chicago, IL.

PEDRONI, M., ORIOL, M., AND MEYER, B. 2007. A framework for describing and comparing courses and curricula. *ACM SIGCSE Bull. 39*, 3, 131–135.

RAND, W. M. 1971. Objective criteria for the evaluation of clustering methods. *J. Amer. Stat. Assoc. 66*, 846–850.

ROWBOTTOM, D. P. 2007. Demystifying threshold concepts. *J. Philosophy Educ. 41*, 2, 263–270.

SCHREIBER, A. 1983. Bemerkungen zur Rolle universeller Ideen im mathematischen Denken. *Mathematica didactica 6*, 65–76.

SCHWILL, A. 1994. Fundamental ideas of computer science. *EATCS Bull. 53*, 274–295.

SCHWILL, A. 2004. Philosophical aspects of fundamental ideas: Ideas and concepts. In *Informatics and Student Assessment—Concepts of Empirical Research and the Standardisation of Measurement in the Area of Didactics of Informatics*, Lecture Notes in Informatics vol. 1, J. Magenheim and S. Schubert, Eds. Köllen Druck + Verlag, Bonn, Germany, 145–157.

THOMAS, M. 2002. Informatische Modellbildung. Modellieren von Modellen als ein zentrales Element der Informatik für den allgemeinbildenden Schulunterricht. Dissertation. Universität Potsdam, Germany.

WARD, J. H. 1963. Hierarchical groupings to optimize an objective function. *J. Amer. Stat. Assoc. 58*, 236–244.

WHITEHEAD, A. N. 1929. *The Aims of Education*. The Free Press, New York, NY.

WURSTHORN, B. 2005. Fundamental concepts of computer science in a Logo-environment. In *Digital Tools for Lifelong Learning. Proceedings of the 10th European Logo Conference (EuroLogo'05)*, G. Gregorczyk, A. Walat, W. Kranas, and M. Borowiecki, Eds. Centre for Informatics and Technology in Education, Warsaw, Poland, 219–227.

Received August 2007; revised January 2008; accepted March 2008