MODESTUM
OPEN ACCESS

EURASIA Journal of Mathematics, Science and Technology Education, 2018, 14(7), 2969-2986
ISSN:1305-8223 (online) 1305-8215 (print)
Research Paper https://doi.org/10.29333/ejmste/91482

Instructional Methods in STEM Education: A Cross-contextual Study

Andreasa Zendler [1*], Cornelia Seitz [1], Dieter Klaudt [1]
[1] University of Education Ludwigsburg, Ludwigsburg, GERMANY

Received 7 January 2018 ▪ Revised 15 March 2018 ▪ Accepted 29 April 2018

ABSTRACT

This study contributes to an integrative view on STEM subjects from an educational point of view. The focus is on the assessment of instructional methods in relation to knowledge processes. By a questionnaire, computer science teachers and mathematics teachers assessed 20 instructional methods in terms of knowledge processes (build, process, apply, transfer, evaluate, and integrate). The findings show that computer science teachers and mathematics teachers differ on the rating of instructional methods. However, the findings also allow a common way of looking at instructional methods by computer science teachers and mathematics teachers. This is an important result for pre- and in-service training programs and for the introduction of computer science as a new school subject.

Keywords: STEM, computer science education, mathematics education, instructional methods, knowledge processes, cross-contextual research

INTRODUCTION

In recent years, national and international initiatives have emerged (Committee on Integrated STEM Education, 2014; Federal Ministry of Education and Research, 2013) to strengthen STEM (Science, Technology, Engineering, Mathematics) subjects and to improve education in STEM sub-jects. MINT is more than a list of individual subjects. In particular, MINT means interdisciplinary lessons, where the lessons are handled in its multi-perspectivity by several subjects. An essential aspect of the MINT initiatives is to build an integrative view at the participating subjects, both from a perspective of learners as well as from a perspective of teachers.

Perspective of learners. From the perspective of learners, an integrative view means that learners have understood related concepts in individual STEM subjects for solving problems (cf. Breiner, Harkness, Johnson, & Koehler, 2012; Johnson, 2013). For example, when building a wind turbine, concepts of various STEM subjects must be combined: from science concepts to capture relationships over real world facts (e.g. wind), from technology concepts to simulate a wind turbine, from engineering concepts to realize artifacts, from mathematics concepts to model real world things (e.g. wind energy) (see **Figure 1**; US Department of Energy, 2007; Windwise Education, 2018; the video of The National Academies of Science Engineering, 2018).

Perspective of teachers. From the perspective of teachers, an integrative view means that teachers offer instructional support, so learners can recognize related concepts in different STEM subjects. In this respect, one of the key questions for an integrative view of STEM subjects is: Which instructional methods most effectively support the act of learning in STEM subjects so that learners can understand the interdisciplinary concepts in STEM subjects?

For science subjects, Treagust (2007) as well as Treagust and Tsui (2014) have drawn up a rough classification of instructional methods. The classification includes seven instructional approaches: demonstration, explanation, questions, representation, analogy and metaphor, cooperative teaching, inductive / deductive approach.

For STEM subjects, STEM programs (see Committee on Integrated STEM Education, 2014, p. 145) propose to favor the instructional methods of problem-based learning and project work: "One implication of this finding is that practices such as engineering design and science inquiry, and instructional approaches like problem- and

✉ zendler@ph-ludwigsburg.de **(*Correspondence)** ✉ Seitz@ph-ludwigsburg.de ✉ klaudt@ph-ludwigsburg.de

Contribution of this paper to the literature

- The results of the study contribute to an integrative view on STEM subjects from an educational point of view.
- The study shows that top rated instructional methods for mathematics and computer science education are problem-oriented learning, and with a lesser degree direct instruction, and project work.
- The results obtained are important for pre- and in-service training programs, especially for countries not yet having a complete teacher training course at university for computer science.

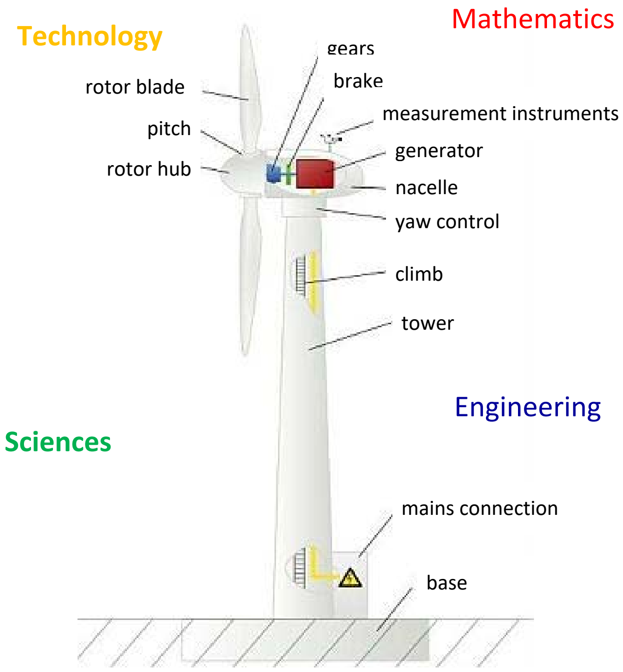


Figure 1. Integrative view at a wind turbine by STEM subjects

project-based learning, may offer special opportunities to support STEM integration when sufficient and intentional instructional support is provided."

Two STEM Subjects: Computer Science and Mathematics

The subject of mathematics was chosen because as a STEM subject it is the subject that is essential in all other STEM subjects: (1) in sciences, mathematics is fundamental for many formulas in physics, chemistry and biology, (2) in technology, mathematics assures processes to optimize artifacts, (3) in engineering, mathematics is the basis for many concepts (e.g. grammar, automata in theoretical computer science).

The subject of computer science was chosen because it is becoming increasingly important as a new subject as curricula show in different countries (European Schoolnet, 2015; Informatics Europe & ACM Europe Working Group, 2013; Ladner & Israel, 2016; The White House, 2016). Ericson (2008) cites the following reasons: (1) the assignment of the subject computer science leads to different career paths, (2) ways of thinking of computer science are important in other disciplines, (3) computer science is important for economy, (4) computer science supports other sciences and establishes connections to those, (5) computer sciences teaches problem solving both in a social and in a scientific way.

The introduction of computer science as a new school subject is a particular challenge: many more computer science teachers are needed than are currently available. One way to solve this problem is to provide specific in-service training programs for teachers that are qualified to teach computer science as a supplementary subject. Teachers who may be eligible for this may be mathematics teachers, because mathematics in part forms the roots of computer science (Becher, 1989). A necessary prerequisite for this is that mathematics and computer science teachers have a similar understanding of instructional methods, which is very important and has to be verified empirically.

Instructional Methods

The wide range of instructional methods is almost incomprehensible. The associated literature describes a broad spectrum of instructional methods ranging from methods of conveying and acquiring knowledge to management methods for games, movement, emotions, groups, health, violence and conflicts. The Center for Teaching and Learning (2018) cites 150 instructional methods, Gugel (2011) more than 2,000 methods including their variations.

There are well-prepared monographs of instructional methods available (e.g. Cruickshank et al., 2011; Petrina, 2006; Petty, 2009). The monograph from Joyce and Weil (2008) is helpful in bringing order to the variety of concepts, with classifications of the instructional methods for *teaching families* (social interaction family, information processing family, personal family, behavioral modification family).

Definitions. Meyer (2002) is a source of a very general definition stating that instructional methods are the forms and procedures by which teachers and students acquire the natural and social reality that surrounds them, taking into account the institutional framework of the school. According to this general definition of instructional method, the idea of what instructional methods mean is very uneven. In addition, a variety of synonyms exist, depending on whether instructional methods are addressed in the context of learning forms, teaching forms or teaching approaches. A narrower definition of method comes from Huber and Hader-Popp (2007, p. 3), which also provides the conceptual starting point for this article: "The word method is understood to mean a clearly defined, conceptually perceivable and independent, if also integrated, component of teaching."

Empirical findings. Numerous empirical findings on the effectiveness of learning are available. In his 800 meta-analyses into which more than 50,000 studies were included Hattie (2009) provides information on the effectiveness of learning with respect to six domains: contributions of the person learning, the parental home, the school, the instructor, the curricula and teaching. In particular, the domain of teaching (Hattie, 2009, Chapters 9 and 10) provides information on the effectiveness of instructional methods/approaches. High effect sizes (d > .50) were demonstrated for microteaching (d = .88), reciprocal teaching (d = .74), feedback (d = .73), problem solving (d = .61), direct instruction (d = .59), mastery learning (d = .58), case study (d = .57), concept mapping (d = .57), peer tutoring (d = .55), cooperative (vs. competitive) learning (d = .54), and interactive instructional videos (d = .52).

Instructional methods for computer science education. As yet no standard reference for computer science education is available which extensively addresses the application of instructional methods for school. The literature contains descriptions on the application of *solving problems* (Koffmann & Brinda, 2003), *group work* (Iron, Alexander, & Alexander, 2004), *rich tasks, concept-mapping* (Hazzan, Lapidot, & Ragonis, 2011), *pattern-oriented instruction* (Muller & Haberman, 2008), *lab-centered instruction* (Titterton, Lewis, & Clancy, 2010), *discovery learning and project teaching* (Hartmann, Näf, & Reichert, 2006), and *visualizations* (Fincher & Petre, 2004).

Instructional methods for mathematics education. For mathematics education a number of good standard references is available which addresses the application of instructional methods: Heddens, Speer, and Brahier (2008), Kidwell and Ackerberg-Hastings (2008), Li, Silver, and Li (2014). A variety of teaching examples with methodical focus are included in practice-oriented journals on mathematics education: e.g. *The Mathematics Educator, The Mathematics Enthusiast, Mathematics Teaching*.

Knowledge Processes Involved in the Act of Learning

The theory from Collins, Brown, and Newman (1989), which has situated learning at its core, reveals four main phases: *modeling, scaffolding, fading,* and *coaching.* Cognitive oriented approaches (Bruner, 1966; Gagné, Wagner, Golas, & Keller, 2004) link instruction to the acquisition and processing of knowledge. They emphasize three knowledge processes in the act of learning: *acquisition of new information, transformation (manipulating knowledge to make it fit to new tasks), evaluation (checking whether the way we have manipulated information is adequate to the task)* (*see* Bruner, 1966, p. 48; Gowda, 2010; Merriam & Caffarella, 2006, p. 46).

The educational literature knows numerous variations relating teaching to learning as an act spread over time and to phases, which can be distinguished during the course of learning (Bruner, 1966; Davis, 2009; Olson, 2007; Petrina, 2006). What all of the variations have in common is that learning (1) has a starting point, (2) a sequential form and (3) a (generally preliminary) end point. Educational literature describes this as the *classic three-step* pattern divided into the steps labeled *entry, work phase* and *graduation.* These three steps have particularly large distinctions in their educational functions and in the knowledge processes of the act of learning.

Particularly in the work phase, important knowledge processes (Bruner, 1966; Gowda, 2010; Merriam & Caffarella, 2006) can be distinguished in the act of learning. This indicates the processes in the acquisition of knowledge (*build, process*), in the transformation of knowledge (*apply, transfer*) and in the evaluation of knowledge (*assess, integrate*). The exact definitions of the knowledge processes are as follows:

- *build:* Acquiring knowledge, new practical and cognitive abilities as well as attitudes;
- *process:* Establishing, deepening, structuring and connecting what has been learned;

- *apply:* Using what has been learned in new tasks corresponding with the framework conditions of the learning situation;
- *transfer:* Using what has been learned in new situations in which the framework conditions differ from those of the learning situation;
- *assess:* Classifying what has been learned in regard to its usefulness, scope, benefits and limits;
- *integrate:* Integrating what has been learned outside of the actual learning situation in connection with one's own knowledge.

RESEARCH QUESTION

For computer science and mathematics education standard references on instructional methods exist (see above). However, it is insufficiently clarified which instructional methods are effective in the act of learning, especially with respect to different knowledge processes. In addition, it is unclear, which instructional methods allow an integrative view on STEM subjects from the perspective of teachers, not even in the context of only two STEM subjects. In this regard, the following research hypothesis will be investigated:

"Computer science teachers differ from mathematics teachers in their ratings of instructional methods with respect to knowledge processes in supporting the act of learning. "

In order to answer the research hypothesis, a procedure is used that is of primary interest in the context of the cross-contextual research paradigm (Berry, Poortinga, Segal, & Dasen; Lerner, Easter Brooks, Mistry, & Weiner, 2003; Saraswathi, 2003). The procedure includes answering three so-called research goals.

(1) *Transport and test goal*: Can the combinations of instructional methods and knowledge processes found in the computer science education context be generalized to the mathematics education context?
(2) *Discover variations goal*: Can other combinations of instructional methods and knowledge processes be found in the mathematics education context that are also important in the computer science education context?
(3) *Assemble and integrate goal*: Can the combinations of instructional methods and knowledge processes identified in the two contexts be integrated to generate a broader perspective that is valid for both contexts?

In the next section, we present the research methods applied, describing the study design and procedures and the data analysis strategy. Then, we give a detailed account of our findings. In the last two sections, we discuss those findings and, finally, draw implications for pre- and in-service training programs.

METHOD

Selection of Instructional Methods

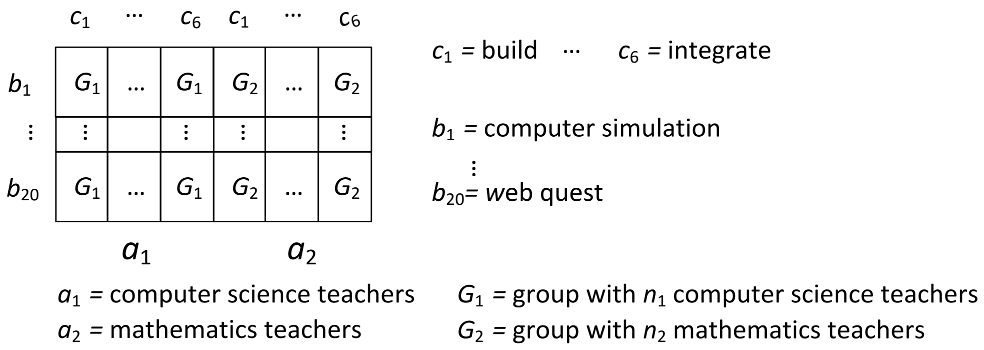
The review of a series of instructional methods manuals (Cruickshank et al., 2011; Davis, 2009; Ginnis, 2001; Joyce & Weil, 2008; Petrina, 2006; Petty, 2009) revealed more than 50 instructional methods to choose from. The review was characterized by the requirement that instructional methods had to pass the muster as being capable of being understood as clearly defined, conceptually perceivable and independent components of the instruction.

The following criteria were applied for the final selection of the instructional methods: (1) The actual application of the instructional methods in computer science and mathematics education, (2) the application of the instructional methods in STEM subjects (sciences, technology, engineering, mathematics) and (3) empirically examined instructional methods. The following 20 instructional methods (in alphabetical order) were selected on the basis of these criteria:

Case study, computer simulation, concept mapping, direct instruction, discovery learning, experiment, jigsaw method, learning at stations, learning by teaching, learning tasks, Leittext method, models method, portfolio method, presentation, problem-based learning, programmed instruction, project work, reciprocal teaching, role-play, and web quest.

Process models for all 20 instructional methods have recently been developed and visualized according to a uniform schema in IDEFO (Integrated Definition for Function Modeling) (Menzel & Mayer, 2005), which make explicit the individual steps in the implementation of the instructional methods in the classroom (Zendler, 2018). **Appendix A-1** contains examples of the process models of case study, computer simulation, and concept mapping.

The hypotheses were tested in a SPF-2•20×6 split-plot design (3-factor design with repeated measures of factors *B* and *C*, see **Figure 2**; Kirk, 1994; Winer, Brown, & Michels, 1991).



	c_1	⋯	c_6	c_1	⋯	c_6
b_1	G_1	…	G_1	G_2	…	G_2
⋮	⋮		⋮	⋮		⋮
b_{20}	G_1	…	G_1	G_2	…	G_2
	a_1			a_2		

c_1 = build ⋯ c_6 = integrate
b_1 = computer simulation
⋮
b_{20}= web quest
a_1 = computer science teachers
a_2 = mathematics teachers
G_1 = group with n_1 computer science teachers
G_2 = group with n_2 mathematics teachers

Figure 2. Experimental design

Independent variables. Factor *A* comprised the *p* = 2 groups surveyed, with factor level a_1 representing group G_1 of n_1 computer science teachers and factor level a_2 representing group G_2 of n_2 mathematics teachers. Factor *B* represented the *q* = *20* instructional methods b_1, ..., b_{20}: case study, computer simulation, concept mapping, direct instruction, discovery learning, experiment, jigsaw method, learning at stations, learning by teaching, learning tasks, Leittext method, portfolio method, presentation, problem-based learning, programmed instruction, project work, reciprocal teaching, role-play, and web quest. Factor C represents the *q* = *6* knowledge processes with factor levels c_1, ..., c_6: *build, process, apply, transfer, assess, and integrate.*

Dependent variable. The dependent variable was the respondents' rating of the instructional methods with respect to the six knowledge processes. Ratings were given on a six-point scale with ratings ranging from 0 ("not significant") to 5 ("very significant").

A power calculation of type II—*N* as a function of power (1–*β*), *Δ*, and *α*—was used to determine the necessary sample size for the 2•20×6 split-plot design (Mueller & Barton, 1989; Mueller, LaVange, Ramey, & Ramey, 1992): With a power (1–*β*) of 0.80, only large effects (*Δ* = 0.80) on the dependent variable being considered significant, and a significance level of *α* = 0.05, a total sample of approximately N^* = 30 (n_1^* = 15 computer science teachers, n_1^* = 15 mathematics teachers) would be required, based on the power computations of Mueller and Barton (1989) or Mueller, LaVange, Ramey, and Ramey (1992) for ε-corrected *F* tests.

Operational Test Hypothesis

Given the study design and the above specification of the independent and dependent variables, the operational hypothesis of the study can be formulated as follows:

"Computer science teachers differ from mathematics teachers in their ratings of instructional methods (case study, computer simulation, concept mapping, direct instruction, discovery learning, experiment, jigsaw method, learning at stations, learning by teaching, learning tasks, Leittext method, models method, portfolio method, presentation, problem-based learning, programmed instruction, project work, reciprocal teaching, role-play, web quest) in supporting the act of learning, operationalized by computer science teachers´ and mathematics teachers' ratings on a six-point scale of the knowledge processes *build, process, apply, transfer, assess and integrate*."

Sample

For the empirical study, 120 computer science teachers in 2014, and 120 mathematics teachers in 2015 at German high schools in the Federal state of Baden-Württemberg were contacted and invited to complete a questionnaire on the usage of instructional methods in computer science and mathematics education. The computer science teachers who sent back the questionnaire, taught computer science in grades 11 and 12/13. On average, they taught about 7.5 years computer science; in addition to the lessons in computer science all computer science teachers taught mathematics. Almost all mathematics teachers who sent back the questionnaire, taught mathematics grades 5 to 12/13. 20 mathematics teachers taught mathematics more than 10 years; in addition to the lessons in mathematics almost all mathematics teachers taught another STEM subject.

Questionnaire

The questionnaire consisted of a short introduction listing the 20 instructional methods and the 6 knowledge processes. The questionnaire was accompanied by a booklet for the computers science teachers (Zendler & Klaudt, 2014) and the mathematics teachers (Zendler, Seitz, & Klaudt, 2015). It describes the 20 instructional methods in accordance with a uniform scheme containing (1) a brief description and explanation, (2) concrete execution steps,

(3) and examples from the relevant literature verifying the application of the instructional method in computer science education and mathematics education, respectively. The questionnaire has been used successfully in several pro-jects and has been validated using the so-called comparative method (Zendler, Klaudt, & Seitz, 2018).

Tasks

The p = 20 instructional methods and the q = 6 knowledge processes were presented in alphabetical order in a matrix with the instructional methods in the rows and the knowledge processes in the columns. Participants were asked to indicate the relevance of each of the 20 × 6 = 120 matrix cells: Each cell represents a combination of an instructional method and a knowledge process and requires an integer from 0 ("not significant") to 5 ("very significant") indicating the relevance of the combination (see **Appendix A-2 Questionnaire**).

Return Rate

To maximize the return rate, we mailed both samples the questionnaires in sealed, personalized envelopes, enclosing a pre-addressed return envelope franked with stamps showing flower designs (see Dillman, 2000 for recommendations on increasing return rates). The return rate for the computer science teachers was 20.0% (n_1 = 24 valid questionnaires of 32 returned questionnaires), which can be considered reasonable for a postal survey (see Vaux & Briggs, 2005). The return rate for the mathematics teachers was 24.25% (n_2 = 29 valid questionnaires of 40 returned questionnaires).

Data Analysis

In analyzing our empirical data, we followed recommended procedures for cross-contextual research (Harkness, van de Vijver, & Mohler, 2003; van de Vijver & Leung, 1997). First, we conducted a three-factor analysis of variance with repeated measures in accordance with the 2•20×6 split-plot design (see Winer, Brown, & Michels, 1991, chapter 7). We conducted a posteriori comparisons of means to test for effects of the $A \bullet B$ interaction and the $A \bullet B \times C$ interaction.

Data analyses were conducted using SPSS 22.0; the power analysis was computed with PASS 13.

RESULTS

Figure 3 visualizes the mean ratings (*see* **Appendix A-3 Data**) obtained from the computer science teachers (a_1) and the mathematics teachers (a_2) for each of the 20 × 6 combinations of instructional methods × knowledge processes the (repeated measures factors $B \times C$). **Figure 3** illustrates that instructional methods in relation to knowledge processes have been rated differently by computer science and mathematics teachers. However, concrete effects are difficult to be identified. Thus, more detailed analyses are necessary in relation to the three cross-contextual research goals (*transport and test goal*, *discover variations goal, assemble and integrate goal).*

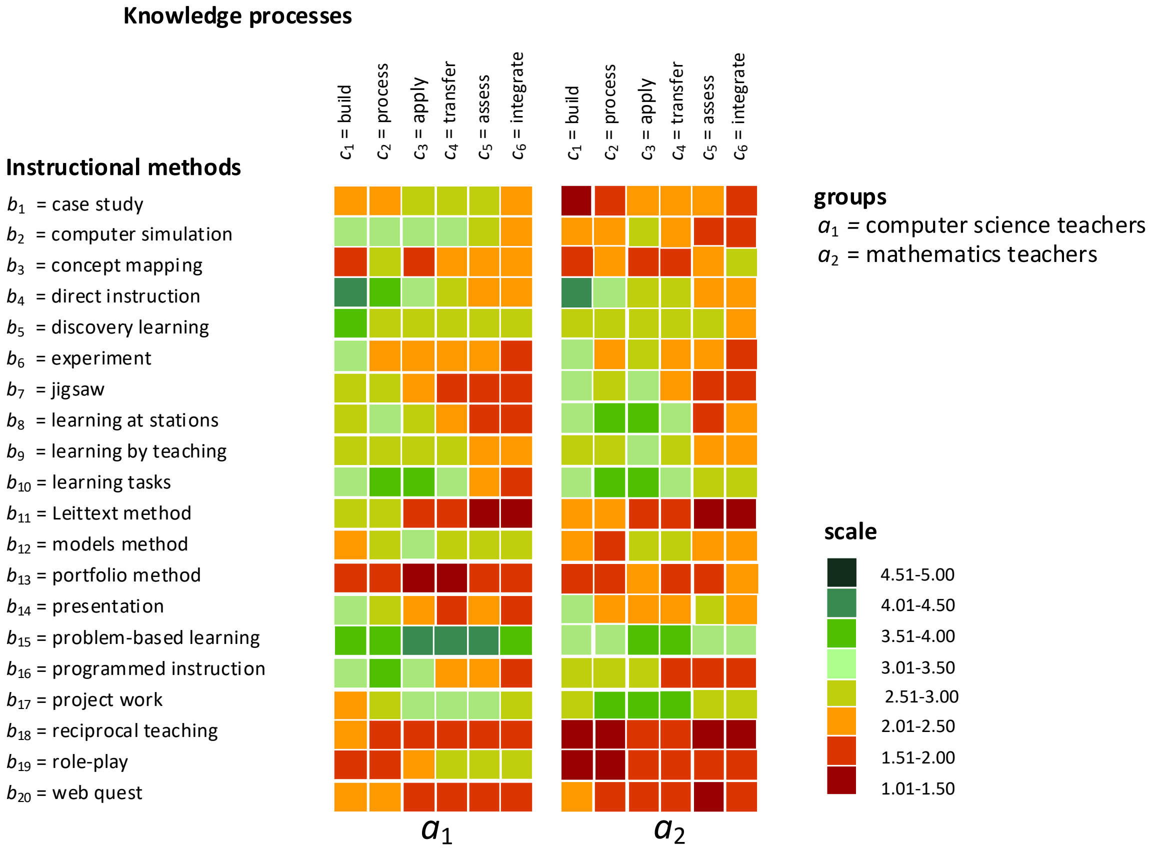


Figure 3. Mean ratings (alphabetically sorted)

Findings for the *Transport and Test Goal*

To examine whether the combinations of instructional methods and knowledge processes identified in the computer science education context can be generalized to the mathematics education context, we formulated three statistical hypotheses, which were tested at the significance level of α = 0.05.

Statistical hypotheses. The three null hypotheses were as follows:

i) the means of the instructional methods μ_1 under a_1 (computer science teachers) and μ_2 under a_2 (mathematics teachers) are equal, such that:

$$H_0\colon \mu_1 = \mu_2$$

ii) the means of the instructional methods $\mu_{1\bullet1}, \mu_{1\bullet2}, \ldots, \mu_{2\bullet20}$ under the 2 • 20 levels of the factor combinations $A \bullet B$ are equal, such that:

$$H_0\colon \mu_{1\bullet1} = \mu_{1\bullet2} = \ldots = \mu_{2\bullet20}$$

iii) the means of the instructional methods $\mu_{1\bullet1\times1}, \mu_{1\bullet1\times2}, \ldots, \mu_{2\bullet20\times6}$ under the 2 • 20×6 levels of the factor combinations $A \bullet B \times C$ are equal, such that:

$$H_0\colon \mu_{1\bullet1\times1} = \mu_{1\bullet1\times2} = \ldots = \mu_{2\bullet20\times6}$$

Testing the statistical assumptions. For an analysis of variance of a split-plot design, the data must satisfy the condition of sphericity. This assumption was tested using Mauchly's W test for sphericity, with the test statistic W being compared to a chi-square distribution to assess the adequacy of the sphericity assumption. The assumption of sphericity was not met for either the instructional methods (W=0.001, $\chi^2_{189} = 369.27$, $p < 0.001$) or the knowledge processes (W=0.372, $\chi^2_{14} = 48.61$, $p < 0.001$) at the α level of 0.05. In the further analyses, we therefore applied the ε correction of degrees of freedom proposed by Huynh and Feldt (1976).

The main effect A (computer science teachers vs. mathematics teachers) was *not* significant at the α level of 0.05 ($F_{1,51}$= 0.54, $p < 0.47$). The corresponding H_0 was therefore *not* rejected: computer science teachers and mathematics teachers did not differ in their *global* ratings of the instructional methods.

Table 1. Analysis of variance

Source of variation	*SS*	*Df*	*MS*	*F*	*p*	*η2*
between subjects						
A	23.61	1	23.61	0.54	< 0.47	< 0.011
error (A)	2233.49	51	43.97			
within subjects						
A • B	200.47	13	15.47	2.12	< 0.02	< 0.040
error (B)	4831.66	661	7.31			
A • B × C	84.40	34	2.38	0.90	< 0.65	< 0.017
error (B × C)	4800.16	1734	2.76			

The interaction effect *A • B* (group • instructional methods) was significant at the α level of 0.05 ($F_{13,\,661} = 2.12$, $p < 0.02$). The corresponding H_0 was therefore rejected: computer science teachers and mathematics teachers differ in their ratings of *individual* instructional methods.

The interaction effect *A • B × C* (group • instructional methods × knowledge processes) was *not* significant at the α level of 0.05 ($F_{34,\,1734} = 0.90$, $p < 0.65$). The corresponding H_0 was therefore *not* rejected: computer science teachers and mathematics teachers did not differ in their ratings of *individual* instructional methods with respect to *individual* knowledge processes.

In summary, our findings for the *transport and test goal* indicate that the combinations of content and process concepts identified by the computer science teachers can not totally be generalized to the mathematics education context.

Findings for the *Discover Variations Goal*

To examine the discover variations goal, two examinations have to be conducted. (1) On the one hand, it must be tested whether computer science and mathematics teachers differ in their ratings of *individual* instructional methods. (2) On the other hand, it must be tested in more detail whether computer science and mathematics teachers differ in their ratings of *individual* instructional methods in relation to *knowledge processes*.

1. Individual comparisons for the *A • B* interactions. To examine whether it was possible to identify combinations of instructional methods in the mathematics education context that are also relevant in the computer science education context, we conducted mean comparisons of the group • instructional methods combinations. These comparisons were conducted using *t* tests to evaluate simple $\overline{AB}$ effects for $p \bullet q \times r$ split-plot designs (Winer, Brown, & Michels, 1991, pp. 535–536), account taken of the ε correction of degrees of freedom ($df = df_{error\ (A)} + df_{error\ (B)} = 51+661=712$; see **Table 1**) for the *t* tests.

Figure 4 visualizes the means and standard errors of the *A • B* interactions for the computer science teachers and the mathematics teachers for each instructional method. The strength of the *simple* $\overline{AB}$ *effects* are visualized and the corresponding *p*-values can be identified by using the *scale of simple* $\overline{AB}$ *effects*. The results of the individual comparisons show that the computer science teachers (a_1) and the mathematics teachers (a_2) differ significantly in their ratings of instructional methods at the α level of 0.05. They differ in the ratings of the following instructional methods:

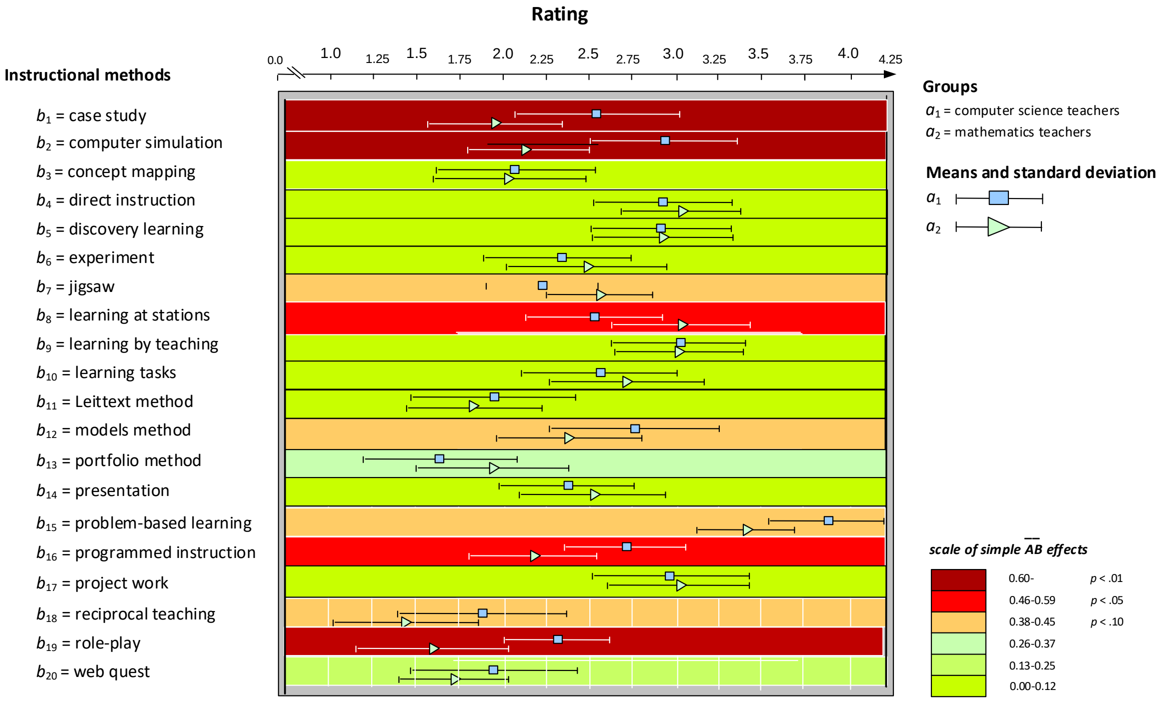


Figure 4. Individual comparisons (*A* • *B* interactions)

- case study (is rated higher by computer science teachers);
- computer simulation (is rated higher by computer science teachers);
- learning at stations (is rated higher by mathematics teachers);
- programmed instruction (is rated higher by computer science teachers);
- role-play (is rated higher by computer science teachers).

2. Individual comparisons for the *A* • *B* × *C* interactions. We next conducted a posteriori pairwise mean comparisons to test which group • Instructional method × knowledge process combinations differed significantly, using 20 × 6 = 120 t tests to evaluate simple $\overline{ABC}$ effects for $p \bullet q \times r$ split-plot designs (Winer, Brown, & Michels, 1991, pp. 535–536), account taken of the ε correction of degrees of freedom ($df = df_{error\ (A)} + df_{error\ (B)} + df_{error\ (C)} + df_{error\ (B \times C)} =$ $51+661+202+1734=2648$ – see **Table 1**) for the t tests. Given the number of t tests that had to be conducted, an adjusted α level was used to determine statistical significance.

Figure 5 shows that computer science teachers and mathematics teachers differ significantly in their ratings of *individual* instructional methods with respect to *individual* knowledge processes. The figure illustrates *simple* $\overline{ABC}$ *effect*s for each *instructional* method with respect to *individual* knowledge processes. The strength of the *simple* $\overline{ABC}$ effects are visualized and the corresponding *p*-values can be identified by using the *scale of simple* $\overline{ABC}$ *effects.*

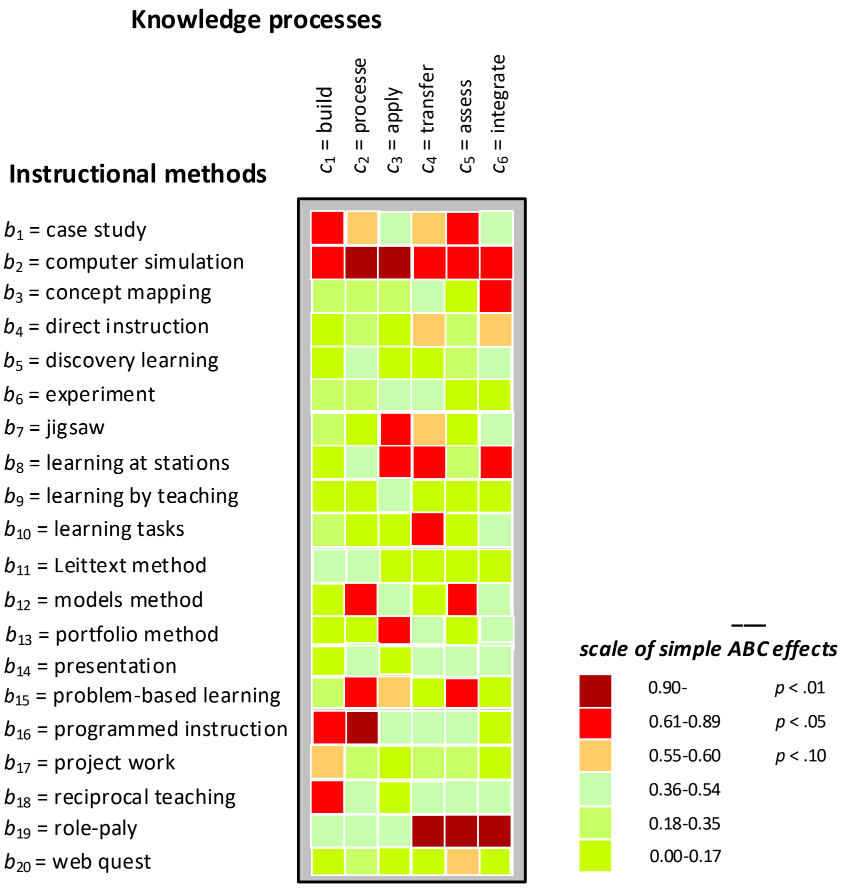


Figure 5. Individual comparisons ($A \bullet B \times C$ interactions)

In sum, our results for the *discover variations goal* show that it is indeed possible to identify combinations of instructional methods and knowledge processes in computer science education context that are also important for the mathematics science education context, as reflected by individual comparisons of the $A \bullet B \times C$ interaction. In particular, significant differences were detected in the two groups' ratings of the relationships of the following instructional methods with respect to *individual* knowledge processes:

- *case study* with respect to *build, assess* (rated higher by computer science teachers)*;*
- computer simulation with respect to build, process, apply, transfer, assess, integrate (rated higher by computer science teachers);
- concept mapping with respect to integrate (rated higher by mathematics teachers);
- jigsaw with respect to apply (rated higher by mathematics teachers);
- learning at stations with respect to apply, transfer, integrate (rated higher by mathematics teachers);
- learning tasks with respect to transfer (rated higher by computer science teachers);
- models method with respect to process, assess (rated higher by computer science teachers);
- portfolio method with respect to apply (rated higher by mathematics teachers);
- problem-based learning with respect to process, assess (rated higher by computer science teachers);
- programmed instruction with respect to build, process (rated higher by computer science teachers);
- reciprocal teaching with respect to build (rated higher by computer science teachers);
- role-play with respect to apply (rated higher by computer science teachers).

Findings for the *Assemble and Integrate Goal*

In order to find answers for the a*ssemble and integrate goal*, the instructional methods are sorted in descending order with respect to the grand means, independently for the computer science teachers and the mathematics teachers (see **Figure 6**). Then instructional methods are highlighted with an asterisk that support at least one

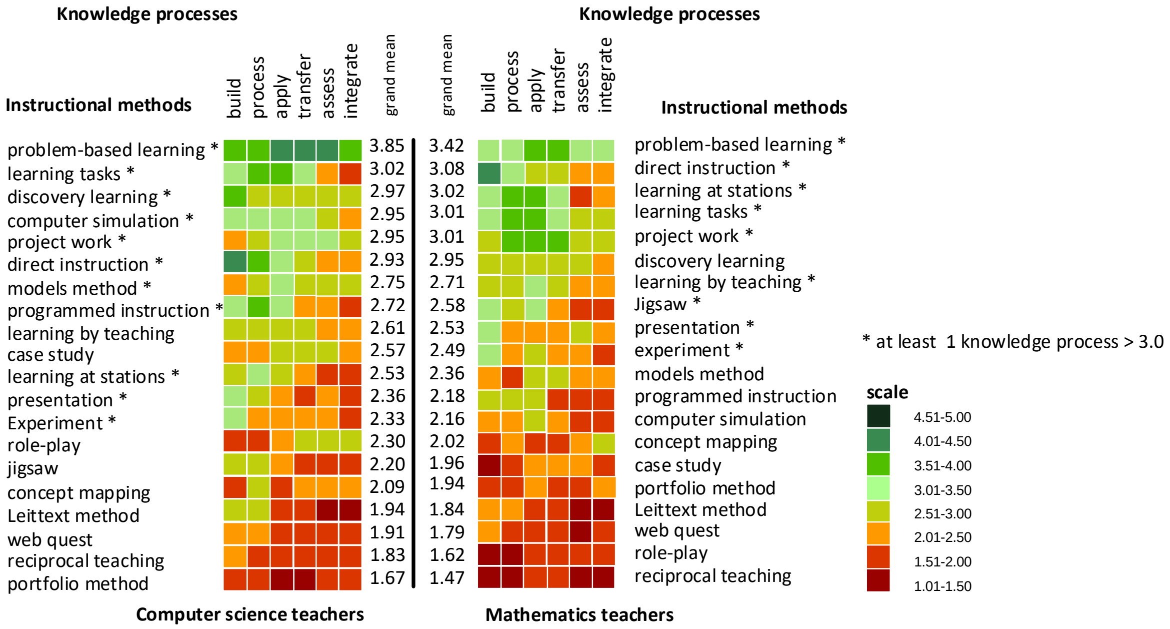


Figure 6. Instructional methods rated by computer science and mathematics teachers (sorted by grand means)

knowledge process very well. The criteria for this is that a knowledge process was rated by the teachers with a value > 3.00.

From the perspective of the computer science teachers, the following eleven methods fulfilled this criterion: problem-based learning, learning tasks, discovery learning, computer simulation, project work, direct instruction, models methods, programmed instruction, learning at stations, presentation, and experiment.

From the perspective of the mathematics teachers, the following nine methods fulfilled this criterion: problem-based learning, direct instruction, learning at stations, learning tasks, project work, learning by teaching, jigsaw, presentation, and experiment.

For the integrative view on instructional methods the criterion that at least one knowledge process was evaluated with a value > 3.00 *both* by computer science *and* by mathematics teachers was used. The following seven instructional methods fulfilled the criterion: problem-based learning, direct instruction, learning at stations, learning tasks, project work, presentation, and experiment.

Figure 7 shows the many similarities in the ratings of instructional methods by computer science and mathematics teachers: problem-based learning is rated highest in terms of almost all knowledge processes, direct instruction is best suited for the knowledge process of *build*, learning tasks are very well suited for the knowledge processes of *process* and *apply*. Project work is appropriate for the knowledge process of *transfer*. With the exception of problem-oriented learning, all instructional methods are more or less inappropriate for knowledge processes of assess and integrate.

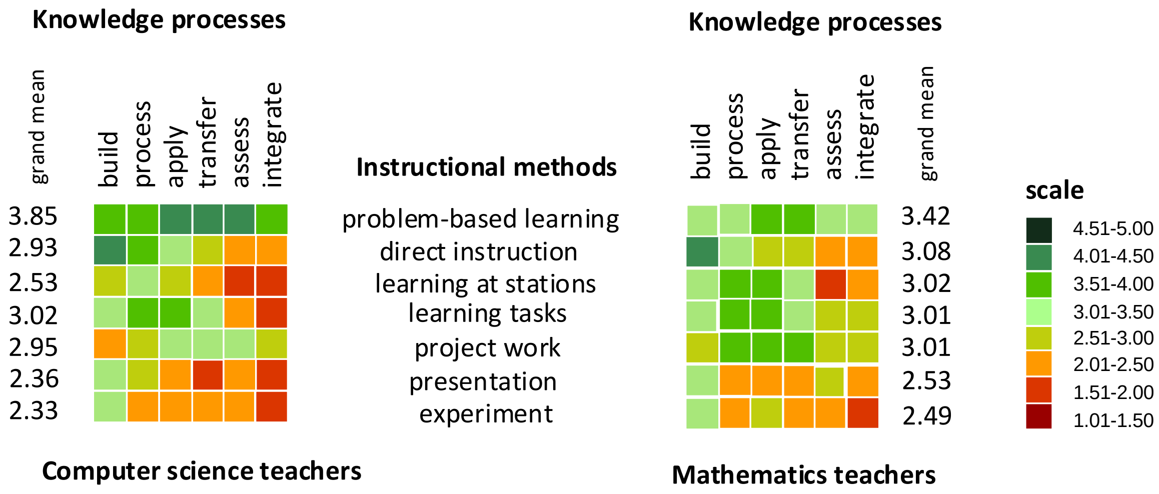


Figure 7. Integrative view on instructional methods

DISCUSSION

It must first be noted that the findings support the research hypothesis formulated in the Introduction that computer science teachers differ from mathematics teachers in their ratings of instructional methods with respect to knowledge processes.

The presented findings on the use of instructional methods in computer science education partly confirm the recommendations made in standard works on the subject of computer science education. This applies to the instructional methods problem-based learning, learning tasks, discovery learning, computer simulation, the project work method and the models method favored in the standard works (Hazzan, Lapidot, & Ragonis, 2011; Iron, Alexander, & Alexander, 2004; Koffmann & Brinda, 2003).

The findings on the application of instructional methods in mathematics education confirm the recommendations made in standard works on the subject of mathematics education. This applies for the instructional methods problem-based learning, learning tasks, learning (at) stations, and project work (Heddens, Speer, & Brahier, 2008; Kidwell & Ackerberg-Hastings, 2008; Li, Silver, & Li, 2014).

The analysis of variance to the *transport and test goal* and the pairwise mean comparisons to the *discover variations goal* showed that computer science and mathematics teachers differ on the ratings of *individual* instructional methods to *individua*l knowledge processes. The greatest differences were found in the rating of computer simulation, case study, programmed instruction, role-play, learning at stations, which the computer science teachers towards the mathematics teachers assessed significantly different. Large differences were also evident in the rating of concept mapping with respect to the knowledge process of *integrate* by mathematics teachers towards computer science teachers. Finally, greater differences were found on the rating of the jigsaw and models method in regard to two knowledge processes.

The results for the *assemble and integrate goal* are very interesting: they allow an integrative view of instructional methods by computer science and mathematics teachers. It was found that both computer science and mathematics teachers top rated problem-oriented learning, and with a lesser degree direct instruction, and project work. The findings support the proposal of the Committee on Integrated STEM Education (2014) to favor these instructional methods, when it comes to the integration of STEM subjects from a teacher's perspective.

The study was conducted with $n_1 = 24$ computer science teachers and $n_2 = 29$ mathematics teachers in the state of Baden-Württemberg who had to make 20 × 6 judgments. The representativeness of the computer science and the mathematics teachers refers to the state of Baden-Württemberg. The results cannot be generalized to other states because of different curricular requirements. Due to the plurality of judgments to be made by the computer science and mathematics teachers maturation effects must be taken into account. For this, the data were analyzed; statistical tests were conducted according to the necessary requirements.

CONCLUSIONS

The results obtained are important for pre- and in-service training programs, especially for countries not yet having a complete teacher training course at university for computer science – as in some states of Germany. The introduction of a new subject in schools represents a particular challenge for teacher training programs. First, the dilemma associated with all new subjects has to be addressed. On the one hand, if teachers are trained before a new subject is introduced, there is a risk that they will not find employment after graduation. On the other hand, a new subject can be successfully introduced only if sufficient numbers of appropriately trained teachers are available. When computer science was first implemented as a compulsory subject in academic-track schools (called Gymnasium) in the southern German state of Bavaria, this dilemma was resolved by providing specific in-service training programs for teachers (called SIGNAL and FLIEG) that qualified to teach computer science as a supplementary subject (Spohrer, 2009). With the obtained result to the *assemble and integrate goal,* educational quality can be achieved when selecting teachers to post-qualify: choosing mathematics teachers, teachers with adequate attitudes on using instructional methods are in focus, even in regard to computer science education.

In follow-up work studies with computer science teachers and teachers, who teach other STEM subjects, should be carried out in order to recruit and post-qualify teachers from a methodological point of view, especially teachers who already possess experiences in instructional methods important to computer science education.

REFERENCES

Becher, T. (1989). *Academic tribes and territories. Intellectual enquiry and the cultures of disciplines*. Milton Keynes: The Society for Research into Higher Education & Open University Press.

Berry, J. W., Poortinga, Y. H., Segal, M. H., & Dasen, P. R. (2002). *Cross-cultural psychology*. Cambridge, UK: University Press.

Breiner, J., Harkness, M., Johnson, C. C., & Koehler, C. (2012). What is STEM? A discussion about conceptions of STEM in education and partnerships. *School Science and Mathematics*, *112*(1), 3–11. https://doi.org/10.1111/j.1949-8594.2011.00109.x

Bruner, J. S. (1966). *The process of education*. Cambridge: Harvard University Press.

Collins, A., Brown, & Newman, S. E. (1989). Cognitive apprenticeship. Teaching the crafts of reading, writing. In L. B. Resnick (Ed.), *Knowing, learning and instruction* (pp. 453–494). Hillsdale: Erlbaum.

Committee on Integrated STEM Education (2014). *STEM integration in K-12 education*. Washington: The National Academies Press.

Cruickshank, D. R., Jenkins, D. B., & Metcalf, K. K. (2011). *The act of teaching*. Boston: McGraw Hill.

Davis, G. B. (2009). *Tools for teaching*. San Francisco: Jossey-Bass.

Dillman, D. A. (2000). *Mail and Internet surveys: the tailored design method*. New York: Wiley.

Ericson, B. (2008). Ensuring exemplary teaching in an essential discipline. Addressing the crisis in *computer science teacher certification*. New York, NY: ACM.

European Schoolnet (2015). Computing our future. Computer programming and coding – Priorities, school curricula and initiatives across Europe. Retrieved on February 24, 2018 from http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0

Federal Ministry of Education and Research (2013). *Perspektive MINT*. Berlin: BMBF.

Fincher, S., & Petre, M. S. (2004). *Computer science education research*. London: Routledge Falmer.

Gagné, R. M., Wagner, W. W., Golas, K., & Keller, J. M. (2004). *Principles of instructional design*. London: Wadsworth Publishing.

Ginnis, P. (2001). *The teacher's toolkit. Classroom achievement*. Carmarthen: Crown House Publishing.

Gowda, N. S. (2010). *Learning and the learner: Insights into the processes of learning and teaching*. Delhi: PHI Learning.

Gugel, G. (2011). *2000 Methoden für Schule und Lehrebildung*. Weinheim: Beltz.

Harkness, J., van de Vijver, F. J. R., & Mohler, P. (2003). *Cross-cultural survey methods*. New York: Wiley.

Hartmann, W., Näf, M., & Reichert, R. (2006). *Informatikunterricht planen und durchführen*. Berlin: Springer.

Hattie, J. (2009). *Visible learning*. New York: Routledge.

Hazzan, O., Lapidot, T., & Ragonis, N. (2011). *Guide to teaching computer science: an activity-based approach.* New York: Springer. https://doi.org/10.1007/978-0-85729-443-2

Heddens, J. W., Speer, W. R., & Brahier, D. J. (2008). *Today's mathematics: Concepts, methods, and classroom activities.* New York: Wiley.

Huber, S.G., & Hader-Popp, S. (2007). Unterrichtsentwicklung durch Methodenvielfalt im Unterricht fördern: das Methodenatelier als schulinterne Fortbildung. In A. Bartz,. J. Fabian, S.G. Huber, Kloft, C. H. Rosenbusch, H. Sassenscheidt (Eds.), *PraxisWissen Schulleitung* (30.31). München: Wolters Kluwer

Huynh, H., & Feldt, L. S. (1976). Estimation of the Box correction for degrees of freedom from sample data in randomised block and split-plot designs. *Journal of Educational Statistics, 1*, 69–82. https://doi.org/10.3102/10769986001001069

Informatics Europe & ACM Europe Working Group (2013). *Informatics education: Europe cannot afford to miss the boat.* Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education. New York: ACM.

Iron, S., Alexander, S., & Alexander, S. (2004). *Improving computer science education*. London: Routledge Chapman & Hall.

Johnson, C. C. (2013). Conceptualizing integrated STEM education. *School Science and Mathematics*, *113*(8), 367–368. https://doi.org/10.1111/ssm.12043

Joyce, B. R., & Weil, M. (2008). *Models of teaching*. New York: Prentice-Hall.

Kidwell, P. A., & Ackerberg-Hastings, A. (2008). *Tools of American mathematics teaching, 1800-2000*. Baltimore: Johns Hopkins University Press.

Kirk, E. (1994). *Experimental design*. Belmont, CA: Wadsworth.

Koffmann, E., & Brinda, T. (2003). Teaching programming and problems solving. In L. Cassel, & R.A. Reis (2003). *Informatics curricula and teaching methods*. (pp. 125-130). Amsterdam: Kluwer Academic Publishers. https://doi.org/10.1007/978-0-387-35619-8_13

Ladner, R., & Israel, M. (2016): 'For All' in 'Computer Science for All. *Communications of the ACM, 9*, 26–28. https://doi.org/10.1145/2971329

Lerner, R. M., Easterbrooks, M., Mistry, J., & Weiner, B. (2003). *Handbook of psychology, Vol. 6: Developmental psychology*. New York: Wiley.

Li, Y., Silver, E. A., & Li, S. (2014). *Transforming mathematics instruction: Multiple approaches and practices*. Berlin: Springer. https://doi.org/10.1007/978-3-319-04993-9

Menzel, C., & Mayer, R. (2005). The IDEF Family of Languages. In P. Bernus, K. Martins, & G. Schmidt (Eds.), *Handbook on Architectures of Information Systems* (pp. 215–250). Berlin: Springer.

Merriam, S. B., & Caffarella, R. S. (2006). *Learning in adulthood: A comprehensive guide*. New York: Bass.

Meyer, H. (2002). Unterrichtsmethoden. In H. Kiper, H. Meyer, & W. Topsch (Eds.), *Einführung in die Schulpädagogik* (pp. 109–121). Berlin: Cornelsen.

Mueller, K. E., & Barton, C. N. (1989). Approximate power for repeated-measures ANOVA lacking sphericity. *Journal of the American Statistical Association*, *84*(406), 549–555. https://doi.org/10.1080/01621459.1989.10478802

Mueller, K. E., LaVange, L. E., Ramey, S. L., & Ramey, C. T. (1992). Power calculations for general linear multivariate models including repeated measures applications. *Journal of the American Statistical Association, 87*(420), 1209–1226. https://doi.org/10.1080/01621459.1992.10476281

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

Olson, D. R. (2007). *Jerome Bruner: The cognitive revolution in educational theory*. New York: Continuum.

Petrina, S. (2006). *Advanced teaching methods for the technology classroom*. New York: Information Science Publishing.

Petty, G. (2009). *Teaching today: a practical guide*. Cheltenham: Nelson Thornes.

Saraswathi, T. S. (2003). *Cross-cultural perspectives in human development: Theory, research and applications*. New York: Sage.

Spohrer, M. (2009). *Konzeption und Analyse neuer Maßnahmen in der Fort- und Weiterbildung von Informatiklehrkräften* (Dissertation). Technische Universität, München.

The Center for Teaching and Learning (2018). *150 teaching methods.* Retrieved on February 24, 2018 from http://teaching.uncc.edu/learning-resources/articles-books/best-practice/instructional-methods/150-teaching-methods

The National Academies of Science Engineering (2018). *STEM Integration in K-12 education*. Retrieved on February 24, 2018 from https://www.nap.edu/catalog/18612/stem-integration-in-k-12-education-status-prospects-and-an

The White House (2016). Obama's computer science for all initiative is gaining Momentum. Retrieved on February 24, 2018 from https://cacm.acm.org/careers/207286-obamas-computer-science-for-all-initiative-is-gaining-momentum/fulltext

Titterton, N., Lewis, C. M., & Clancy, M. J. (2010). Experiences with lab-centric instruction. *Computer Science Education, 20,* 2, 97–102. https://doi.org/10.1080/08993408.2010.486256

Treagust, D. (2007). General instructional methods and strategies. In S. K. Abell & N. G. Lederman (Eds.), *Handbook of research on science education* (pp. 373–441). Mahwah, NJ: Lawrence Erlbaum Associates.

Treagust, D., & Tsui, C.-Y. (2014). *General instructional methods and strategies.* In N. G. Lederman & S. K. Abell (Eds.), *Handbook of research on science education II* (pp. 303–320). Mahwah, NJ: Lawrence Erlbaum Associates.

US Department of Energy (2007). *Building the basic PVC wind turbine*. Saint Paul: Kidwind Project.

van de Vijver, F., & Leung, K. (1997). *Methods and data analysis for cross-cultural research*. Thousand Oaks, CA: Sage.

Vaux, A., & Briggs, C. S. (2006). Conducting mail and internet surveys. In F. T. L. & J.T. Austin (Eds.), *The psychology research handbook* (pp. 186–209). London: Sage. https://doi.org/10.4135/9781412976626.n13

Windwise Education (2018). *Transforming the energy of wind into powerful minds*. Retrieved on February 24, 2018 from https://www.scientrific.com.au/PDFs/Lesson-8.pdf

Winer, B. J., Brown, D. R., & Michels, K. M. (1991). *Statistical principles in experimental design*. Boston, MA: McGraw-Hill.

Zendler, A. (Ed.) (2018). *Unterrichtsmethoden in MINT-Fächern*. Berlin: Springer.

Zendler, A., & Klaudt, D. (2014). *The Booklet I. Instructional methods to computer science education*. Berlin: epubli.

Zendler, A., Klaudt, D., & Seitz, C. *(2018). Instructional methods in STEM and English subjects: A validation study.* International Journal of Research in Education and Science *(accepted).*

Zendler, A., Seitz, C., & Klaudt, D. (2015). *The Booklet II. Instructional methods to mathematics education*. Berlin: epubli.

APPENDIX

A-1 Process Models of Instructional Methods (Three Examples)

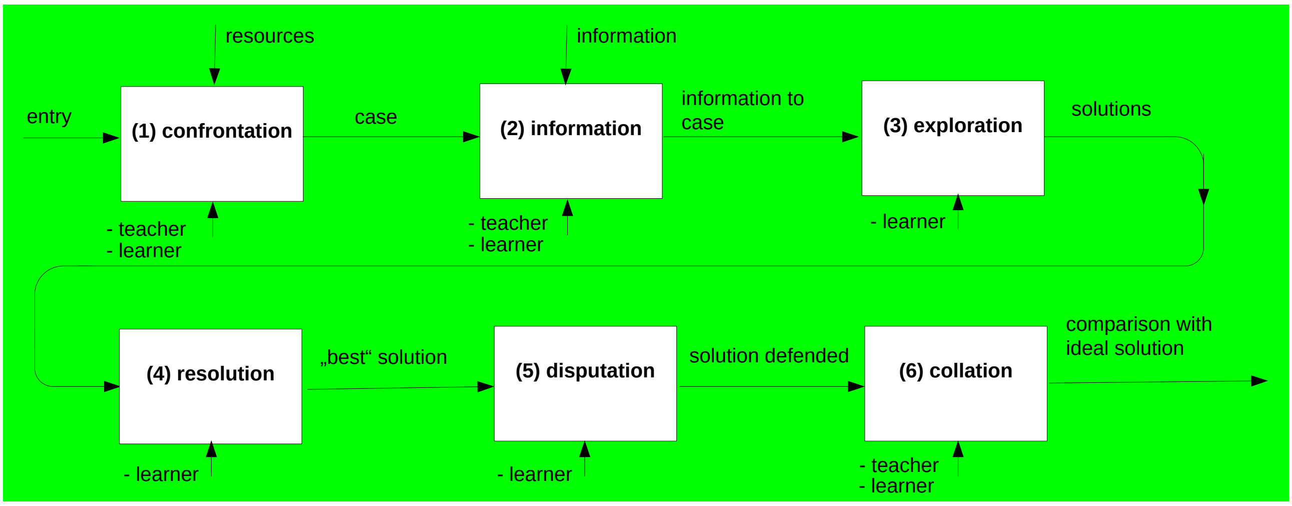


Figure A-1. Case study

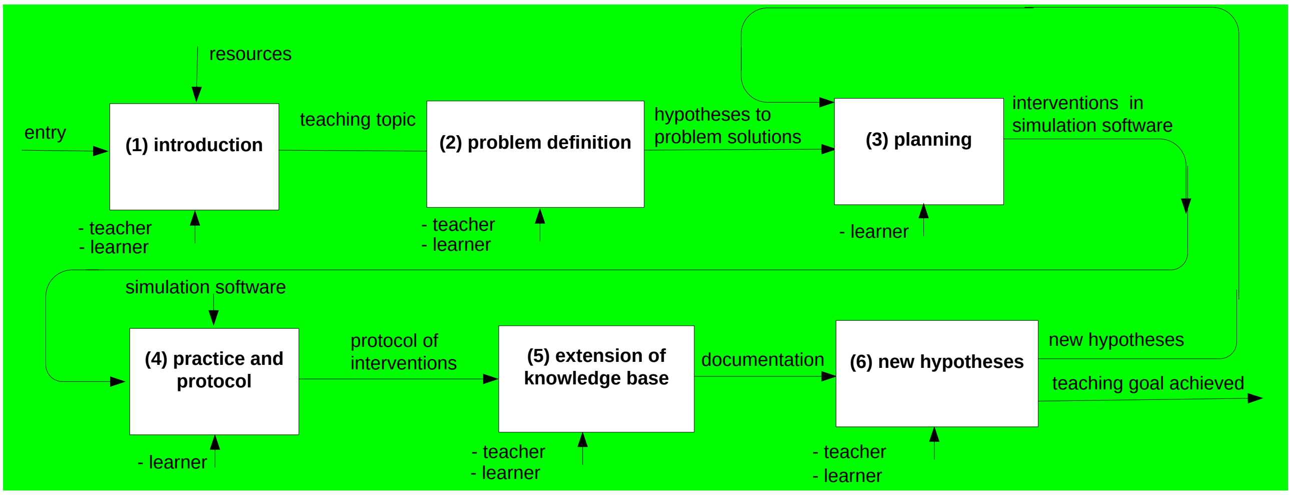


Figure A-2. Computer simulation

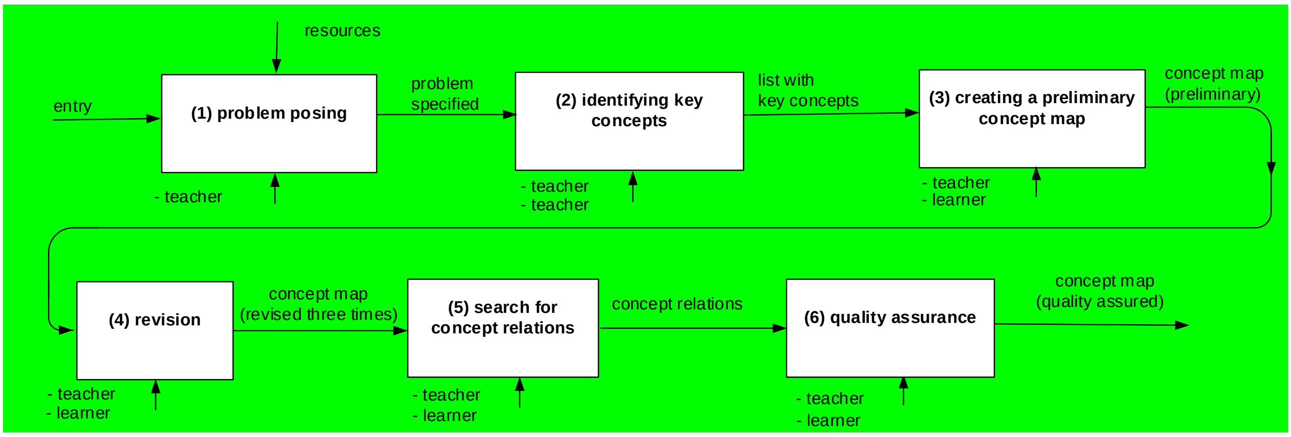


Figure A-3. Concept mapping

APPENDIX

A-2 Questionnaire

Please evaluate:

The act of learning through instructional methods.

Please rate each cell on a scale of 0 to 5 (only whole numbers).

It is important that you provide 6 ratings per row.

	Knowledge processes (Explanations, see Booklet) **Instructional methods** (Explanations, see Booklet)	build	process	apply	transfer	assess	integrate
1	case study						
2	computer simulation						
3	concept mapping						
4	direct Instruction						
5	discovery learning						
6	experiment						
7	jigsaw						
8	learning at stations						
9	learning by teaching						
10	learning tasks						
11	Leittext method						
12	models method						
13	portfolio method						
14	presentation						
15	problem-based learning						
16	programmed instruction						
17	project work						
18	reciprocal teaching						
19	role-play						
20	web quest						

APPENDIX

A-3 Data

Knowledge processes **Instructional methods**	c_1 = build	c_2 = process	c_3 = apply	c_4 = transfer	c_5 = assess	c_6 = integrate	*grand means*
b_1 = case study	2.25	2.21	2.83	2.83	2.88	2.42	2.57
b_2 = computer simulation	3.21	3.00	3.13	3.17	2.75	2.38	2.94
b_3 = concept mapping	1.92	2.33	1.71	2.08	2.29	2.21	2.09
b_4 = direct instruction	4.29	3.58	3.08	2.58	2.00	2.08	2.94
b_5 = discovery learning	3.71	2.67	2.83	2.92	2.92	2.92	2.99
b_6 = experiment	3.29	2.13	2.50	2.04	2.08	1.92	2.33
b_7 = jigsaw	2.96	2.92	2.25	1.71	1.79	1.58	2.20
b_8 = learning at stations	2.96	3.17	2.96	2.42	1.88	1.88	2.54
b_9 = learning by teaching	2.75	2.83	2.88	2.67	2.38	2.17	2.61
b_{10} = learning taks	3.00	3.92	3.67	3.38	2.17	1.96	3.01
b_{11} = Leittext method	2.83	2.63	1.92	1.83	1.25	1.25	1.95
b_{12} = models method	2.13	2.67	3.00	2.96	2.92	2.88	2.76
b_{13} = portfolio method	1.71	1.92	1.42	1.38	1.92	1.67	1.67
b_{14} = presentation	3.08	2.83	2.33	1.92	2.00	1.96	2.35
b_{15} = problem-based learning	3.67	3.75	4.21	4.00	4.00	3.54	3.86
b_{16} = programmed instruction	3.42	3.63	3.13	2.42	2.04	1.71	2.72
b_{17} = project work	2.25	2.92	3.33	3.29	3.13	2.75	2.94
b_{18} = reciprocal teaching	2.08	1.96	1.79	1.83	1.79	1.75	1.87
b_{19} = role-play	1.67	1.79	2.29	2.79	2.79	2.58	2.32
b_{20} = web quest	2.33	2.17	1.67	1.79	1.92	1.71	1.93
grand means	2.78	2.75	2.65	2.50	2.34	2.16	***2.53***

a_1 = computer science teachers

	c_1 = build	c_2 = process	c_3 = apply	c_4 = transfer	c_5 = assess	c_6 = integrate	*grand means*
b_1 = case study	1.41	1.62	2.38	2.28	2.17	1.90	1.96
b_2 = computer simulation	2.41	2.03	2.55	2.45	2.00	1.52	2.16
b_3 = concept mapping	1.59	2.07	1.52	1.72	2.41	2.83	2.02
b_4 = direct instruction	4.31	3.52	2.97	2.97	2.21	2.48	3.07
b_5 = discovery learning	3.83	3.00	2.90	2.76	2.66	2.55	2.95
b_6 = experiment	3.48	2.31	2.76	2.38	2.14	1.86	2.49
b_7 = jigsaw	3.24	2.93	3.07	2.31	1.93	2.00	2.58
b_8 = learning at stations	3.07	3.62	3.76	3.21	1.97	2.48	3.02
b_9 = learning by teaching	2.83	2.93	3.38	2.72	2.28	2.14	2.71
b_{10} = learning taks	3.17	3.86	3.66	2.72	2.28	2.38	3.01
b_{11} = Leittext method	2.34	2.24	1.97	1.83	1.38	1.28	1.84
b_{12} = models method	2.03	2.00	2.55	2.86	2.28	2.45	2.36
b_{13} = portfolio method	1.69	1.97	2.14	1.90	1.90	2.07	1.94
b_{14} = presentation	3.14	2.45	2.38	2.28	2.52	2.45	2.53
b_{15} = problem-based learning	3.48	3.03	3.62	3.90	3.17	3.31	3.42
b_{16} = programmed instruction	2.59	2.72	2.59	2.00	1.59	1.59	2.18
b_{17} = project work	2.83	3.10	3.28	3.10	2.97	2.79	3.01
b_{18} = reciprocal teaching	1.38	1.45	1.69	1.66	1.38	1.28	1.47
b_{19} = role-play	1.21	1.31	1.93	1.76	1.86	1.66	1.62
b_{20} = web quest	2.38	1.93	1.79	1.69	1.34	1.62	1.79
grand means	2.62	2.51	2 64	2.42	2.12	2.13	***2.41***

a_2 = mathematics teachers

Figure A-4. Mean ratings (n_1=24; n_2=29)

http://www.ejmste.com