Informatics in Education, 2023, Vol. 22, No. 1, 141–160

DOI: 10.15388/infedu.2023.07

Promoting Student Competencies in Informatics Education by Combining Semantic Waves and Algorithmic Thinking

Frauke RITTER, Bernhard STANDL
Karlsruhe University of Education
Bismarckstraße 10, 73131 Karlsruhe, Germany
e-mail: frauke.ritter@ph-karlsruhe.de, bernhard.standl@ph-karlsruhe.de

Received: January 2022

Abstract. We live in a digital age, not least accelerated by the COVID-19 pandemic. It is all the more important in our society that students learn and master the key competence of algorithmic thinking to understand the informatics concepts behind every digital phenomena and thus is able to actively shape the future. For this to be successful, concepts must be identified that can convey this key competence to all students in such a way that algorithmic thinking is integrated in the subject of informatics -beyond a pure programming course. Furthermore, based on the Legitimation Code Theory, semantic waves provide a way to develop and review lesson plans. Therefore, we planned a workshop, that follow the phases of a semantic wave addressing algorithmic problems using a block-based programming language. Considering this, we suggest the so-called SWAT concept (**S**emantic **W**ave **A**lgorithmic **T**hinking concept), which is carried out and analyzed in a workshop with students. The workshop was carried out in online format in an 8th grade of a high school during a coronavirus lockdown. The level of algorithmic thinking was measured using a pretest and posttest both in the treatment group and in a control group and with the help of the approximate adjusted fractional Bayes factors for testing informative hypotheses statistically and through a reductive, qualitative content analysis of the students' work results (worksheets and created programs) evaluated. The semantic wave concept was measured using several cognitive load ratings of the students during the workshop and also statistically evaluated with the approximate adjusted fractional Bayes factors for testing informative hypotheses, as well as a qualitative content analysis of the worksheets. Results of this pilot study provide first insights, that the SWAT-concept can be used in combination of unplugged and plugged parts.

Keywords: Algorithmic Thinking, Semantic wave, Block-based programming.

1. Introduction

In our increasingly digitized world, it is of particular importance that all students are taught the skills of computational thinking so that in the future, they will not only become users of digital technologies, but will be able to determine, understand, implement and develop these technologies that make our lives more and more. Tissenbaum and Ottenbreit-Leftwich (2020) emphasizes that today's students have to learn the *language of programming* so that in 2030 a broad social base will understand the fundamental concepts of informatics -especially computational thinking as a central problem-solving competence.

Informatics still has not been introduced as a compulsory education subject in all German federal states Schwarz *et al.* (2021). In the course of the digital transformation, however, this is likely to change in the foreseeable future. Considering this, increasing didactic focus must be placed on informatics as a general education subject, both in the training of future informatics teachers and in the already existing elective courses. The goal should be there to teach all students computational thinking as a core competency in informatics. The SWAT concept (**S**emantic **W**ave **A**lgorithmic **T**hinking concept) we developed is intended to support this goal of promoting algorithmic thinking with block-based programming languages. In the following, the scientific background and related work that led to the development of the SWAT concept is explained first, followed by the concept itself and the resulting research questions and hypotheses. In the fourth section, the methods of the study are presented. This includes the setting of the study with the workshop design, the description of the participants as well as the survey instruments and analysis methods. Finally, the results are presented and discussed in a conclusion with a view to further research.

2. Background to the SWAT Concept and Related Work

In computer programming education, the identification of strategies for teaching problem-solving skills and individual learning experiences receives traditionally considerable attention (Papavlasopoulou *et al.*, 2019). In particular, teaching and learning strategies in the context of the constructionism is of importance in algorithmic problem-solving (Csizmadia *et al.*, 2019). Hence, the analysis of an efficient design for teaching activities with a focus on algorithmic problem-solving processes or algorithmic thinking is an essential area of research in informatics education (Standl, 2017). Considering this, the SWAT concept as suggested in this paper includes: a *semantic wave* structure, the promotion and communication of algorithmic thinking (**S**emantic **W**ave **A**lgorithmic **T**hinking concept) and the use of block-based programming languages for solving an algorithmic problem.

2.1. *Semantic Wave*

The *Legitimation Code Theory* Maton (2013) describes the *semantic wave* to model teaching processes. Fig. 1 of Waite *et al.* (2019) shows a traversing *semantic wave*. The *semantic gravity* (SG) and *semantic density (SD)* are essential parts of it. Semantic gravity is defined: *Semantic gravity (SG) refers to the degree to which meaning relates*

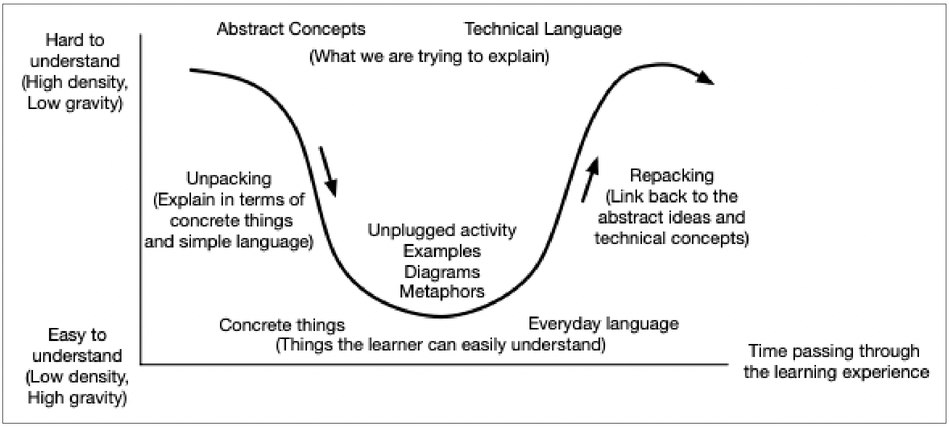


Fig. 1. Traversing a *semantic wave* by Waite *et al.* (2019).

to its context. (...) The stronger the semantic gravity (SG+), the more meaning is dependent on its context; the weaker the semantic gravity (SG-), the less dependent meaning is on its context. The Semantic density: *Semantic density (SD) refers to the degree of condensation of meaning within socio-cultural practices, whether these comprise symbols, terms, concepts, phrases, expressions, gestures, clothing, etc. (...) The stronger the semantic density (SD+), the more meanings are condensed within practices; the weaker the semantic density (SD-), the fewer meanings are condensed.* Maton (2013) describes the course of a *semantic wave* in the context of a biology and a history lesson. The lessons begin with a low SG and a high SD; Waite *et al.* (2019) describe it as *Experts can easily understand*,a phase of unpacking follows in which the SG increases and the SD decreases.

Waite *et al.* (2019) further applied the *semantic wave* approach to a sequence of informatics lessons in which the concept of the algorithm is unplugged in elementary school and conclude that the approach in the case study is effective. They encourage further investigations into how the acquisition of computational thinking and general IT skills can be described, understood, and ultimately controlled with *semantic waves*. In a more recent study by Curzon *et al.* (2020), they also research the application for two more unplugged informatics lessons and conclude that the *semantic wave* concept is suitable for evaluating lesson sequences. Fincher *et al.* (2020) describe in their article on notional machines in informatics lessons the point in time at which notional machines come into play in a semantic wave. According to Fincher *et al.* (2020), this notional machine would strengthen the semantic gravity and reduce the semantic density and at the same time focus the cognitive load on the essentials.

2.2. *Algorithmic Thinking*

Wing (2008) considered *Algorithmic Thinking* as a part of the concepts of computational thinking, which ultimately goes back to Papert (1982). In recent decades, many approaches and attempts to define computational thinking have been discussed but a common definition of *Computational Thinking* is still missing. Cansu and Cansu (2019) give an extensive overview of the different definitions of computational thinking. For Fraillon *et al.* (2019) and the ICILS 2018 Eickelmann *et al.* (2019), Computational Thinking is: *Computational thinking refers to an individual's ability to recognize aspects of real-world problems which are appropriate for computational formulation and to evaluate and develop algorithmic solutions to those problems so that the solutions could be operationalized with a computer.*

This broad definition also includes algorithmic thinking, as Standl (2017) based on the remarks by Wing (2008), Barr and Stephenson (2011) argued, can be understood as part of computational thinking. Sadykova and Il'bahtin (2020) define algorithmic thinking as: *Algorithmic thinking is a system of thinking methods that is necessary to build a sequence of obtaining intermediate results, planning the structure of actions and its implementation, leading to the achievements of the goal.* In this article, focusing on computational thinking, algorithmic thinking is understood as part of computational thinking in the sense of the authors mentioned above and examined by Standl (2017) based on everyday problems. Concluding the problem-solving process in the context of computational thinking can be described in a three-stage process: 1. Describe, abstract and decompose the problem, 2. Design the algorithm, 3. Test the solution. This procedure also fulfills Sadykove's definition of algorithmic thinking when applied to plugged problems to be solved.

2.3. *Block-based Programming*

Block-based programming languages have been used in informatics classes for many years and, due to the lack of syntax problems, can also be introduced in early grades, in which the students had no previous programming experience at all, as was the goal of the developers of the block-based programming language Scratch 2009, for example, Resnick *et al.* (2009). In the years that have passed since then, many block-based programming languages with a wide variety of application areas have emerged and have been investigated. Park and Shin (2019), for example, compare MIT AppInventor2 with Scratch3 and come to the conclusion that 'large' programs, in particular, i.e. those with a lot of code, strongly promote computational thinking.

3. The SWAT Concept

This research pursues the general goal of educating students in the digitized future to become competent, self-determined citizens, who are proficient in computational thinking and, in particular, algorithmic thinking. Our approach systematically examines the didactic effects of block-based programming languages on students; the starting points and guidelines are Standl (2017) three-step approach to conveying problem-solving processes and the *semantic wave* investigated by Waite *et al.* (2019) and by Curzon *et al.*

(2020). The aim was to develop a teaching-learning structure that can be applied to conveying algorithmic thinking as a sub-area of computational thinking in teaching situations with a block-based programming language – the SWAT concept which is structured as follows:

1. Methodologically, the workshop follows a *semantic wave* (Maton, 2013) and the three-step scheme adapted from Standl (2017) for conveying algorithmic thinking.
2. In terms of content, an algorithmic question is addressed in the workshop and solved with a block-based programming language for secondary school students.

Hence, the SWAT concept is framed in the methodical approach of the *semantic wave* and the CT problem-solving process. Based on this, an algorithmic problem is to be solved with a block-based programming language. This approach is shown in Fig. 2.

Considering the conceptional idea of SWAT approach different phases in informatics lessons are to be made visible and, easier to plan and evaluate. On the one hand, how from Curzon *et al.* (2020) found that the teachers plan a lesson well with semantic waves and also are able to reflect on it later. On the other hand, however, a focus is placed on algorithmic problem-solving processes based on Standl (2017). An individual differentiation is also given in this concept at each work step through different possible programming solutions. Thus, our suggested concept for the methodical process of a workshop with a special focus on algorithmic thinking should convey this key competence to the learners in a sustainable manner and at the same time offer the teachers a good opportunity to plan their workshops and evaluate them afterwards.

Since the SWAT concept follows the *semantic wave* Maton (2013) on the one hand and the scheme by Standl (2017) to promote algorithmic thinking, on the other hand, the following research questions arise, assuming, students of lower secondary level take part in a workshop driven by the SWAT concept:

- To what degree do students perceive the phases of the *semantic wave*?
- Does the students' algorithmic thinking competence develop?

The suggested SWAT concept is a new methodical approach in informatics didactics and the use of the *semantic wave* in an informatics lesson with plugged parts has not yet

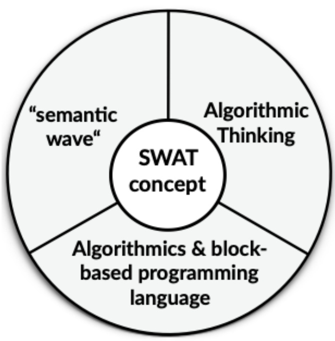


Fig. 2. SWAT concept.

been examined in detail. Thus, this study examines whether the methodical approach of the *semantic wave* is used at all can be implemented with computer (plugged) content in such a way that the students also perceive it, as well as, whether the competence of algorithmic thinking is promoted in the students. To operationalize the research questions this study is designed as an explanatory, empirical individual case study that follows the mixed-method approach; both qualitative and quantitative data is collected and evaluated. According to the case study approach as suggested by Robert K. Yin (2018), this is a "single-case" study with "multiple units of analysis". We will examine the following hypotheses:

- **Hypothesis 1 (H1)**: The students from lower secondary level school classes perceive the phases of a *semantic wave* during the workshop.
- **Hypothesis 2 (H2):** The SWAT concept promotes algorithmic thinking in students.

4. Methods

In this section, the structure and implementation of the workshop based on the SWAT concept and the participants are described first, followed by an explanation of the survey instruments and analysis methods.

4.1. *Setting*

The SWAT concept was applied in a workshop for 8th-grade students. Based on Curzon *et al.* (2020) and Waite *et al.* (2019), we designed a 90-minute workshop, that followed a *semantic wave*, where the content focuses on the left-hand rule of the Pledge algorithm for finding one's way out of a maze through algorithmic problem-solving. Didactically it implements the block-based programming language Scratch. Fig. 3 shows the schematic course of the workshop.

At the beginning of the workshop (*Semantic profile phase: 1 Signalling and Concept Introduction*), the teacher gives a short input on the problem and methodical approach (*Lesson Plan Step: 1 Explanation of the problem and 1a of the procedure (T)*) – where *semantic gravity* is low and *semantic density* is high. The students then use worksheets to develop the left-hand-rule algorithm or the Pledge algorithm in a three-stage problem-solving process (*Semantic profile phases: 2 Concrete activity (AT), phase 3 and 4: Staged return coupled with concrete activity (AT)*). In each of these three problem-solving phases, the students are instructed to follow the three steps of algorithmic problem-solving as described above. The problem is first described and solved unplugged and then solved and analyzed plugged (*Lesson Plan Steps 2–4: Solving problem a and b*). Fig. 4 shows the worksheets for phases 2–4 with increasingly difficult mazes.

In phase 2 (*Concrete activity*) the problem is formulated in simple colloquial language and the maze is kept so simple that, on the one hand, the left-hand rule is sufficient to solve the problem and on the other hand, the maze is also very simple. This ensures

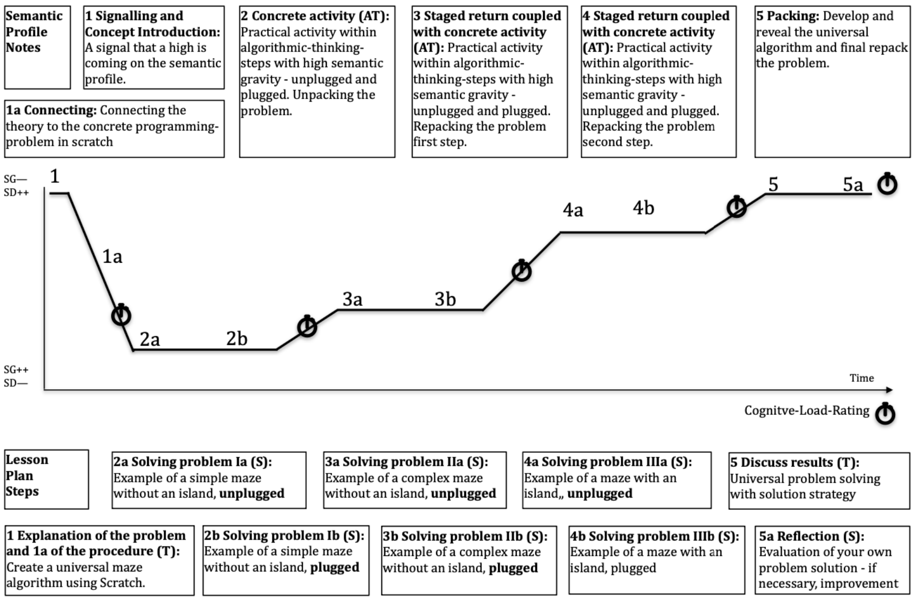
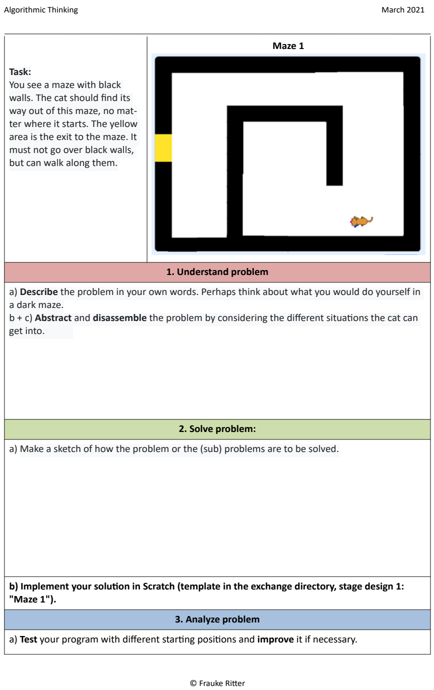


Fig. 3. Workshop Design with Semantic Profile Notes and Lesson Plan Steps
(AT = Algorithmic Thinking Steps, T = Teacher, S = Students).

a high *semantic gravity* and a low *semantic density*. In phases 3 and 4 (*Staged return coupled with concrete activity (AT)*) the *semantic density* increases and the *semantic gravity* decreases – the mazes become more complex. In phase 4 the left-hand rule is no longer sufficient; this is where the Pledge algorithm has to be found (*Staged return coupled with concrete activity (AT)*). In the last phase 5 *Packing*, the results are compared and discussed (*Lesson Plan Steps: Discuss results (T) and Reflection (S)*) – the *semantic gravity* is low again and the *semantic density* is high again.

Due to the coronavirus pandemic, the entire investigation took place online. As shown below, Smith *et al.* (2007) *treatment fidelity* categories (*adherence*, *quantity*, *quality* and *process*) were taken into account when conducting the study.

The workshop itself was carried out using a video conference, online worksheets and the online Scratch programming environment. All students had to solve the same tasks and had the same amount of time available for them (*treatment fidelity – quantity*). The students had the competence to create programs with Scratch and were familiar with elementary process structures and variables. The workshop was structured in such a way that individual work steps were guided by a separate online worksheet form. After each work step, the students submitted their interim results digitally (*treatment fidelity – adherence*). The entire setting of the lesson also ensures the *treatment fidelity* element *process*. During the workshop, parallel to the cognitive load (Fig. 3), the students rated their personal effort in the previous phase, so the *treatment fidelity* element *quality* was measured by this additional compliance item (means increase steadily from 2 to 5, in which 1 is the worst and 7 being the best on the Likert scale, standard deviation decreases; compliance increases throughout the workshop.).

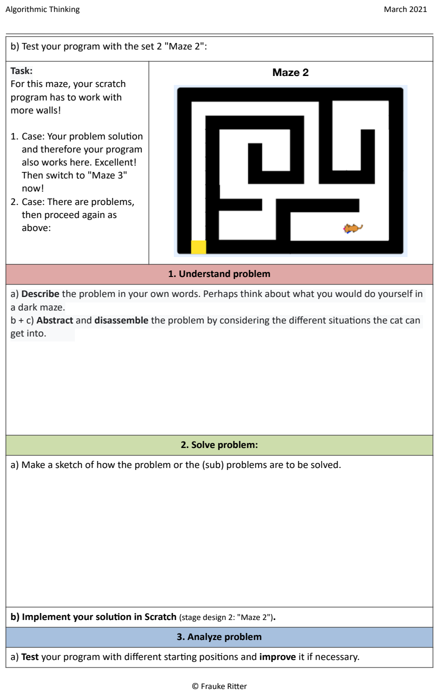


Algorithmic Thinking March 2021

Task: You see a maze with black walls. The cat should find its way out of this maze, no matter where it starts. The yellow area is the exit to the maze. It must not go over black walls, but can walk along them.	Maze 1
1. Understand problem	
a) **Describe** the problem in your own words. Perhaps think about what you would do yourself in a dark maze. b + c) **Abstract** and **disassemble** the problem by considering the different situations the cat can get into.	
2. Solve problem:	
a) Make a sketch of how the problem or the (sub) problems are to be solved.	
b) Implement your solution in Scratch (template in the exchange directory, stage design 1: "Maze 1").	
3. Analyze problem	
a) **Test** your program with different starting positions and **improve** it if necessary.	

© Frauke Ritter

(a) Task for Maze 1

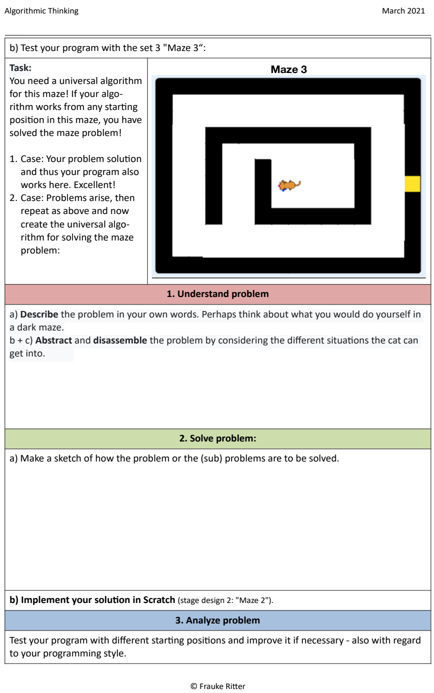


Algorithmic Thinking March 2021

b) Test your program with the set 2 "Maze 2":	
Task: For this maze, your scratch program has to work with more walls! 1. Case: Your problem solution and therefore your program also works here. Excellent! Then switch to "Maze 3" now! 2. Case: There are problems, then proceed again as above:	**Maze 2**
1. Understand problem	
a) **Describe** the problem in your own words. Perhaps think about what you would do yourself in a dark maze. b + c) **Abstract** and **disassemble** the problem by considering the different situations the cat can get into.	
2. Solve problem:	
a) Make a sketch of how the problem or the (sub) problems are to be solved.	
b) Implement your solution in Scratch (stage design 2: "Maze 2").	
3. Analyze problem	
a) **Test** your program with different starting positions and **improve** it if necessary.	

© Frauke Ritter

(b) Task for Maze 2



Algorithmic Thinking March 2021

b) Test your program with the set 3 "Maze 3":	
Task: You need a universal algorithm for this maze! If your algorithm works from any starting position in this maze, you have solved the maze problem! 1. Case: Your problem solution and thus your program also works here. Excellent! 2. Case: Problems arise, then repeat as above and now create the universal algorithm for solving the maze problem:	**Maze 3**
1. Understand problem	
a) **Describe** the problem in your own words. Perhaps think about what you would do yourself in a dark maze. b + c) **Abstract** and **disassemble** the problem by considering the different situations the cat can get into.	
2. Solve problem:	
a) Make a sketch of how the problem or the (sub) problems are to be solved.	
b) Implement your solution in Scratch (stage design 2: "Maze 2").	
3. Analyze problem	
Test your program with different starting positions and improve it if necessary - also with regard to your programming style.	

© Frauke Ritter

(c) Task for Maze 3

Fig. 4. Worksheet Design.

4.2. *Participants*

The treatment group consisted of 19 students as part of the informatics class in 8th grade. The students were 13–14 years old and both boys and girls participated.

The students had already completed an introductory course in programming with Scratch in the informatics class of the 8th grade. Elementary process structures were known, the variable concept as well. Other topics in the 8th grade of the informatics curriculum include the basics of media technology (storing data, image processing, data exchange in networks), information and communication systems (construction of networks with a focus on the Internet), and technical informatics (automated processes in everyday life, modeling of machines) thus, no further programming content.

The control group consisted of 25 students of the informatics course in the 9th grade and thus 14–15 years old – also boys and girls. Although the control group was one year older, these students had not gained any further programming experience (in Scratch) in their informatics courses. Up to the time of the study, the curriculum for informatics in the 9th grade included coding and cryptology (data exchange in networks, data protection and security), databases and social networks (construction of networks with a focus on the Internet) as well as simple website creation. According to the curriculum, the programming skills will only be deepened again in the 10th grade. So, it can be assumed that the prior knowledge of the two groups of students is roughly the same – even if grade 9 students are a year older.

The students (both the treatment group and the control group) completed an algorithmic thinking test (section 4.3) using online forms – a pretest and posttest. Both groups of students had identical test conditions in terms of questions, form and time (*treatment fidelity – quantity*). By comparing the pretest and posttest, the learning gain from the workshop can be measured for the treatment group, since the control group did not experience any intervention. The pretest and posttest were used here as control values for learning gains.

4.3. *Instruments*

4.3.1. *Instruments for Hypothesis 1: The Students from Lower Secondary Level Perceive the Phases of a Semantic Wave During the Workshop*

Subjective cognitive load has long been set in teaching research. The theory states that the working memory is limited and added to the intrinsic cognitive load (ICL) and the extraneous cognitive load (ECL). The ICL becomes a relationship for the complexity of the gestural learning tasks of the learner. The ECL is caused by the presentation and design of the learning material. If learners have to use their cognitive efforts to search for information that is important for the task or to repress redundant information, these are extraneous processes. Since working memory resources are limited, it is important to keep the ECL low if there is more capacity for the ICL. As Paas *et al.* (1994) describes, single item rating belongs to the learner's ICL, as well as being subjective. Klepsch *et al.* (2017) developed and validated a central rating scale in which two items ask about the

ICL of the learners and three items ask about the ECL so that it can be assumed that this test is reliable and valid. Skulmowski and Rey (2020) came to the conclusion that the questionnaire by Klepsch *et al.* (2017) is for learning settings in which digital interactive learning media are used. To measure the perception of the *semantic wave*, our validated questionnaire was carried out by Klepsch *et al.* (2017) because, on the one hand, programming with Scratch can be viewed as a digital interactive learning medium and on the other hand, it is clear that a high *semantic density* (technical language formulations) and low *semantic gravity* (abstract concepts) require a higher *intrinsic cognitive load* of the students and a low *semantic density* (everyday language) and high *semantic gravity* (real-world examples) need a lower *intrinsic cognitive load* from the students. In correspondence, the ECL access was also heard to listen to the perception of the *semantic wave* by the students. Fig. 3 shows the Cognitive-Rating-Times during the Workshop. In addition to the questionnaire the worksheets were used in the analysis.

4.3.2. *Instruments for Hypothesis 2: The SWAT Concept Promotes Algorithmic Thinking in Students*

This hypothesis was examined using an Algorithmic Thinking questionnaire (pretest and posttest with control group). Román-González (2015) developed and validated a questionnaire to measure algorithmic thinking, so that it can be assumed the test is reliable (Cronbach alpha pretest and posttest: 0.8 – good) and valid. Based on this questionnaire, the items that were relevant for the present study (sequences, loops, conditionals) were translated from English into German. The test was filled in with varying disruptors for students before and after the workshop. In addition, a control group from the 9th year completed the questionnaire twice at the same time interval but without going through the workshop. In addition to the questionnaire, the worksheets and the Scratch programs were used in the analysis.

4.4. *Analysis*

4.4.1. *Analysis for Hypothesis 1: The Students from the Lower Secondary Level Perceive the Phases of a Semantic Wave During the Workshop*

The *intrinsic cognitive load* (ICL) was measured by the questionnaire of Klepsch *et al.* (2017) using two items on a 7-point Likert scale at five different points in time (see Fig. 3). 1 on the Likert scale means that the ICL was low and therefore the task was simple (*semantic density* low and *semantic gravity* high); accordingly, a value of 7 means that the ICL is high and the task is therefore very complex (*semantic density* high and *semantic gravity* low). The higher the value on the Likert scale, the more demanding a task is cognitively rated. The ECL was measured using three items on a 7-point Likert scale. 1 means that the ECL was low and thus the presentation of the task is simple, correspondingly a value of 7 means that the ECL is high and thus the presentation of the task is very confusing. The ICL construct is identified using two items, the ECL using three items, from which the mean values per student were formed.

The ICL and ECL construct measurements are statistically descriptive evaluated and the inferring statistics were calculated with the help of the approximate adjusted fractional Bayes factors for testing informative hypotheses (Hoijtink *et al.* (2019)) using the within-repeated measurement ANOVA. The effect sizes were calculated using the standardized Cohen's d.

The worksheets were evaluated using a reductive, qualitative content analysis based on the interpretative paradigm according to Mayring and Fenzl (2019). The analysis focuses on hypothesis H1 (*The students from the lower secondary level perceive the phases of a semantic wave during the workshop.*). The question was differentiated from the theory by structuring the material in the analysis units (phases 2–4) and examining the *semantic density* and *semantic gravity* in the students' answers using a developed category system. This category system was first developed, coded on the material and then discussed and revised by the team before a student trained on the revised category system coded the material. The intercoder agreement in MAXQDA was 79.79% with a segment overlap of 95%. The coding was then revised again by adding statements such as "I don't know" to the category system using a separate code: *Semantic wave: not assignable*. In addition, text passages were coded for the same length and forgotten passages of text from each other were taken over. So that after the revision there was a 91.89% intercoder match in MAXQDA with a 95% segment overlap.

4.4.2. *Analysis for Hypothesis 2: The SWAT Concept Promotes Algorithmic Thinking in Students*

The algorithmic thinking measurements collected from the students (treatment and control group) through Román-González (2015) questionnaire were statistically descriptively evaluated, and the inferring statistics were also calculated with the help of the approximate adjusted fractional Bayes factors for testing informative hypotheses (Hoijtink *et al.* (2019)) using the within-repeated measurement ANOVA. The effect sizes were calculated also using the standardized Cohen's d.

The worksheets were evaluated for this Hypothesis using a reductive, qualitative content analysis based on the interpretive paradigm according Mayring and Fenzl (2019), except that the analysis was directed towards hypothesis H2 (*The SWAT concept promotes algorithmic thinking in students.*). In this case, the research hypothesis was differentiated from the theory of algorithmic thinking by structuring the material in the analysis units (phases 2–4 of the semantic wave) and using the developed coding categories *Understand Problem (UP)* and *Solve Problem (SP)* with the respective characteristics weak, medium and strong. This category system was also initially developed, coded on the material, discussed and revised in a team, and counter-coded with the help of a trained student. The intercoder agreement in MAXQDA was 78.31% with a segment overlap of 95%. As a result, the coding was revised by adding statements such as "I don't know." as an anchor example for the category *Understand Problem – weak* and "Feeling along the wall" both in *Understand Problem* and *Solve Problem* with the expression weak. Again, text passages were coded for the same length, and code mixups were corrected. So that after the revision there was a 93.47% intercoder match in MAXQDA with a 95% segment overlap.

The Scratch programs were also analyzed using a reductive, qualitative content analysis based on the interpretative paradigm according to Mayring and Fenzl (2019). The research hypothesis was differentiated from the theory of algorithmic thinking and individual categories of Moreno-León and Robles (2015) by also structuring the material in the analysis units (phases 2–4 of the semantic wave) and using the developed coding categories *Flow Control* and *Logic* with the respective characteristics weak, medium and strong.

Since there was exactly one Scratch program (coding unit) for each problem and precisely one code for the two categories *Flow Control* and *Logic* assigned to each problem, the intercoder agreement was determined using the category "Existence. The code in the document checked.". After the first intercoding comparison, this resulted in a percentage agreement of 87.61%. Then three incorrectly coded *Logic* codings were improved together. The final intercoder agreement then resulted in a value of 94.5% in MAXQDA.

5. Results

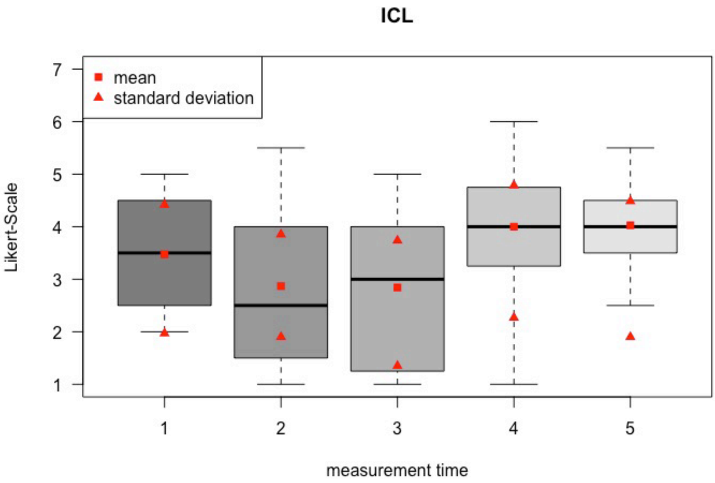
5.1. *Results for Hypothesis 1: The Students from the Lower Secondary Level Perceive the Phases of a Semantic Wave During the Workshop*

Fig. 5(a) shows a wave-like, descriptive course of the measurement of the intrinsic cognitive load (ICL). The informative hypothesis was: $ICL1 > ICL2 < ICL3 < ICL4 < ICL5$, where ICL1 means ICL measurement at time 1 and correspondingly ICL2 to ICL5 (see Fig. 3). The null hypothesis was $ICL1= ICL2= ICL3= ICL4= ICL5$. To test these hypotheses, we used approximate adjusted fractional Bayes factors for testing informative hypotheses using the within-repeated measurement ANOVA. The result of the calculations is strong relative evidence for the informative hypothesis (200,000,000 times more likely).

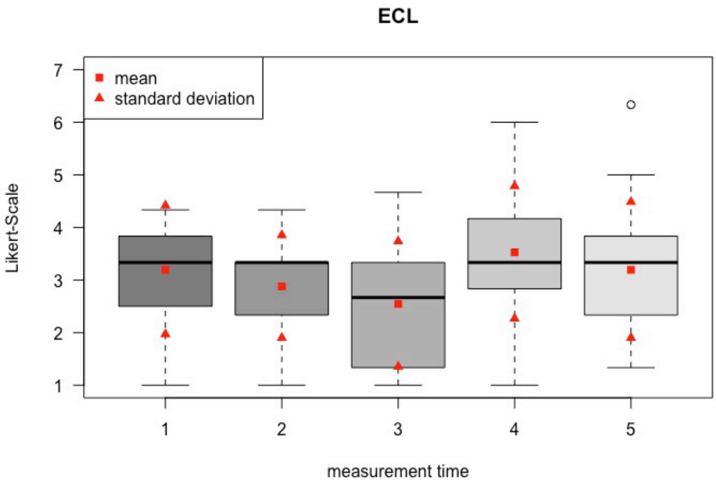
The effect sizes underline this result since there are no differences between the mean values at time points 2 and 3 at other time points, but the differences between time points 1, 4 and 5 are moderate to large (see Table 1).

The informative hypothesis for the measurement of the extraneous cognitive load was: $ECL1 > ECL2 > ECL3 > ECL4 > ECL5$, where ECL1 means ECL measurement at time 1 and correspondingly ECL2 to ECL5 (see Fig. 3). The null hypothesis was $ECL1= ECL2= ECL3= ECL4= ECL5$. To test these hypotheses, we also used approximate adjusted fractional Bayes factors for testing informative hypotheses using the within-repeated measurement ANOVA. The results of the calculation showed no evidence for the hypothesis as shown in Fig. 5(b). There is no downward trend in the measurement of the extraneous cognitive load, also the effect sizes for the different descriptions of statistics can be interpreted as negligible.

Fig. 6 shows a document comparison diagram between the worksheets Maze1Maze3 and the individual students (numbers 1–19). It can be seen that the students in phase 2 predominantly used formulations corresponding to the *semantic wave* (12 of the 19 stu-



(a)



(b)

Fig. 5. Cognitive Load.. (a) Intrinsic Cognitive Load, (b) Extraneous Cognitive Load.

Table 1

Cohen's d for ICL

	ICL1	ICL2	ICL3	ICL4	ICL6
ICL1	-	0.52 moderate	0.41 small	-	-
ICL2	-	-	-	-	-
ICL3	-	-	-	-	-
ICL4	0.36 small	0.72 moderate	0.65 moderate	-	-
ICL5	0.41 small	0.85 strong	0.72 moderate	-	-

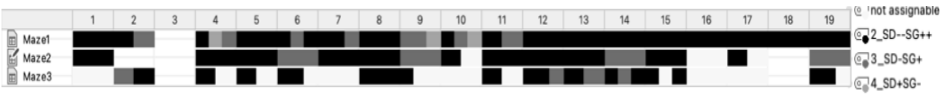


Fig. 6. Occurrence of the *semantic wave* categories in the forms weak, medium and strong in phases 2–4 of the *semantic wave* (maze 1–3) per student (number 1–19).

dents; Code *2_SD-SG* ++). These are utterances such as “She can feel her way ahead on the wall or turn if she has run into it.” In phases 3 and 4, however, no significant change in the formulations according to the categories *semantic density* and *semantic gravity* can be seen, as only 4 out of 19 students express themselves accordingly. Instead, the non-categorized utterances such as “I ran out of time” or “I don’t know” increase sharply in phase 4 (12 of 19 students).

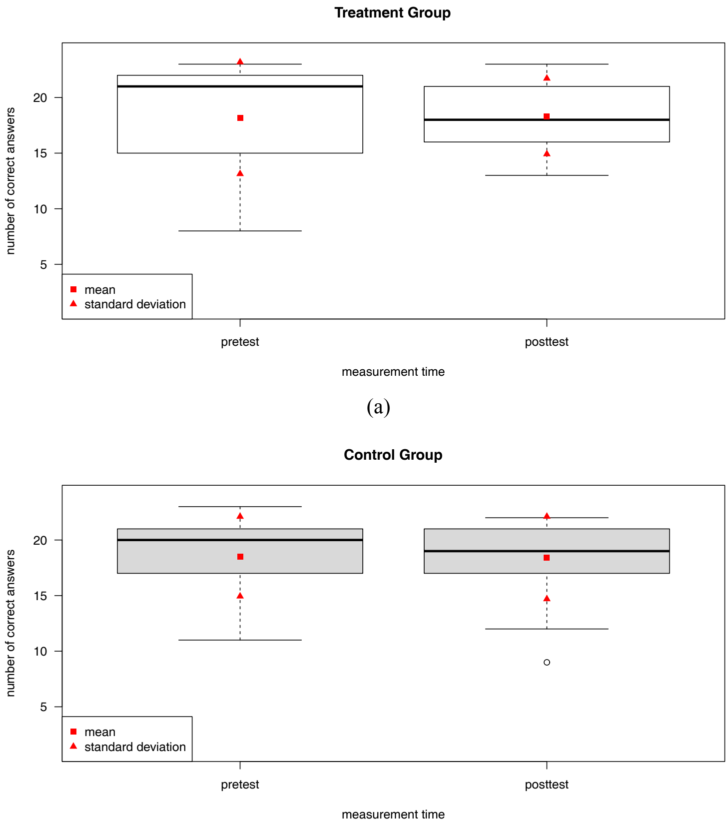
In summary, the following result arises for the first hypothesis: The results of the ICL measurement show a course that corresponds to the course of a *semantic wave* – the students thus perceived the course of the *semantic wave* accordingly. Based on the qualitative analysis, however, the students’ answers could only be assigned accordingly in phase 2; in phases 3 and 4, the students also formulated predominantly with high *semantic density* and low *semantic gravity*. The proportion of codings that cannot be assigned increases in phase 4 to almost half of the codings.

5.2. *Results for Hypothesis 2: The SWAT Concept Promotes Algorithmic Thinking in Students*

In Fig. 7 (b), it can be seen that in the control group there were almost no differences between the pretest and posttest, only the median fell. This is also the case with the treatment group but here the lower variance is reduced (Fig. 7 (a)).

The informative hypothesis was: $TG\ pretest = CG\ pretest = CG\ posttest < TG\ posttest$ with *TG: treatment group* and *CG: control group*. The null hypothesis was $TG\ pretest = CG\ pretest = CG\ posttest = TG\ posttest$. To test these hypotheses, we also used approximate adjusted fractional Bayes factors for testing informative hypotheses using the within-repeated measurement ANOVA. The results of the calculation showed no evidence for the hypothesis, also the effect sizes for the different descriptions of statistics can be interpreted as negligible.

The content analysis initially shows that in both the *Understand Problem (UP)* and *Solve Problem (SP)* categories on all three worksheets, the value weak occurs most frequently (see Table 2). These are in the *Solve Problem* category in phase 4, answers such as: “Arm touches black, run, nose touches black, turn right, the arm does not touch black, turn left”, which in this case are not sufficiently effective for the island maze (maze 3) and remain at the example level. Accordingly, the utterances coded weakly are for phases 2 and 3. Comparing this with the implementation in phase 2, a difference is noticeable: 12 of the 19 students completely solved task 1 (*Logic* strong), although the answers in the categories *Understand Problem (UP)* and *Solve Problem (SP)* were mostly only weak or medium. The first maze was implemented 8 times with a variant of



(a)

(b)

Fig. 7. Results of the Algorithmic Thinking Test. (a) Treatment group, (b) Control group.

the left-hand rule and three times with a variant of the right-hand rule. This also explains the percentage increase in the category *Understand Problem (UP)* and *Solve Problem (SP)* and the percentage decrease in these weak characteristics in phase 3 (Maze 2) (see Table 2). In the *Logic* category, 11 students remained constant with their category level (medium to strong), although task 2 was more difficult – thus they improved their performance. 3 of the 12 students in the category *Logic* with level strong in phase 2, only got medium in phase 3, as the right-hand rule in the second maze does not lead to the exit of the maze. In phase 4, the formulation of the problem-understanding and problem-solving no longer shows any increase, instead 8 students stated that they did not have enough time to completely solve the problem. Four of them used the left-hand rule, which, however, failed with the island maze – a total of 6 students had this problem. In the end, 2 students completely solved the task and 12 had problems with the island, 3 could not find a solution, which is why in the category *Logic* only two students have the rating strong, 12 the rating medium and 3 weak. The coding *Flow Control* remained medium to strong throughout the workshop, which can be explained by the previous knowledge of the students of this course.

Table 2

Occurrence of the codes in the worksheets and the programs of the *semantic wave* phases 2–4 (Maze 1–3) in absolute and relative numbers

Codes	Maze 1		Maze 2		Maze 3	
	absolute	relative	absolute	relative	absolute	relative
Understand Problem weak	17	85.0%	9	56.3%	7	61.5%
Understand Problem medium	2	10.0 %	1	6.3%	4	30.8%
Understand Problem strong	1	5.0 %	6	37.5%	1	7.7%
Sum Understand Problem	20	100.0 %	16	100.0%	12	100.0%
Solve Problem weak	7	41.2%	9	52.9%	8	53.3%
Solve Problem medium	8	47.1%	4	23.5%	3	20.0%
Solve Problem strong	2	11.8%	4	23.5%	5	26.7%
Sum Solve Problem	17	100.0 %	17	100.0%	16	100.0%
Logic weak	2	10.5%	2	11.8%	3	17.6%
Logic medium	5	26.3%	7	41.2%	12	70.6%
Logic strong	12	63.2%	8	47.1%	2	11.8%
Sum Logic	19	100.0 %	17	100.0%	17	100.0%
Flow Control weak	0	0.0%	0	0.0%	0	0.0%
Flow Control medium	12	64.7%	11	64.7%	11	64.7%
Flow Control strong	7	35.3%	6	35.3%	6	35.3%
Sum Flow Control	19	100.0 %	17	100.0%	17	100.0%

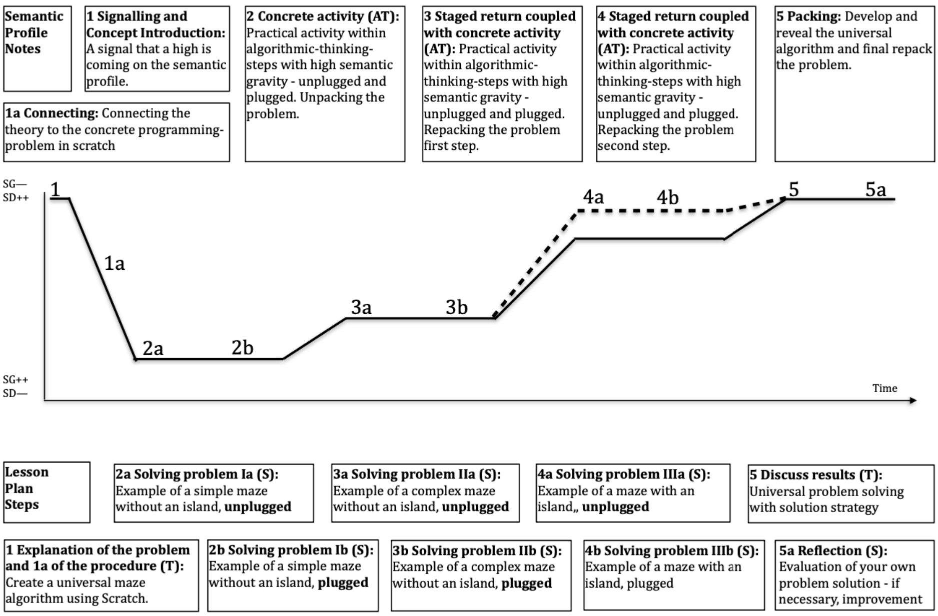


Fig. 8. The actual course of the workshop (dashed line).

Thus, the following result for the second hypothesis can be formulated: Statistically, no significant increase in algorithmic thinking could be determined. The results of the qualitative analysis of the worksheets and the created programs show that between phases 2 and 3 in particular, an increase in the students' algorithmic thinking can be determined – since the categories *Understand Problem (UP)* and *Solve Problem (SP)* were processed better and the students, despite the difficult task during the implementation did not decrease in the category *Logic*, but mostly remained at a high level. In addition, the coding of phase 4 in the categories *Understand* and *Solve Problem*, as well as in the category *Logic*, shows that the students had time problems to solve the last maze well. With the help of these results, the actual course of the workshop can be described as a *semantic wave*, but the wave was more like the following Fig. 8 shows.

Thus, some students had problems in Lesson Plan Steps 4a respectively 4b (*Solving problem 4a and 4b*) and thus, in the semantic wave phase 4 *Stage return coupled with concrete activity (AT)*, the *semantic gravity* decreases, and the *semantic density* increases more in this area than originally assumed.

6. Discussion and Further Research

The statistical evaluation of the students' *Intrinsic Cognitive Load Rating* during the workshop showed that the students very likely perceived the phases of the *semantic wave* according to the SWAT concept. Thus, the construction of the pilot workshop according to the developed SWAT concept with plugged and unplugged parts within a *semantic wave*, was well implemented and should be tested further in subsequent studies.

Statistically, no significant increase in algorithmic thinking could be identified. This could be because algorithmic thinking cannot be statistically significantly increased immediately after 90 minutes, but positive observations can be described from the qualitative analysis, as explained above. According to this pilot study, it can be seen that the approach to promoting algorithmic thinking with the help of a workshop based on the SWAT concept is also promising but should be examined over a longer period than just one workshop. This is the next thing to investigate.

The qualitative content analysis of the worksheets showed, in all investigations of this study that the students had problems both using more abstract technical language and generalizing examples (*SD + SG-*), as well as formulating the steps of algorithmic thinking unplugged (*Understand Problem* and *Solve Problem*). Hence, this should be focussed on in further investigations, or, in any case, should be given much greater weight in the classroom than the pure implementation of tasks, as the problem-solving process only becomes clear here.

Since some students did not have enough time or had problems with maze 3, it is essential to pay attention to the issues concerning duration and degree of difficulty in the following investigations. In this context, the last task should also be easier but more open, so that all students can complete the last phases, complete the semantic wave and higher-performing students are also promoted in a differentiated manner.

The results of the ICL measurement show a course that corresponds to the course of a *semantic wave*. The ECL values do not correspond with the expected course. This could be due to the pandemic-related online implementation of the workshop and, thus, to the shape of the online worksheets. These do not allow a look back or a holistic view of the problem during processing. Therefore, the workshop should be tested in person in the following studies or with analog worksheets or both.

The presented study is a pilot study and subsequent research will evaluate our developed SWAT concept in other students of different ages and with different content. The next research step planned is a 4-part workshop in which algorithmic content will be conveyed. Due to the longer duration of the intervention, it will be examined whether a significant increase in learning can be determined among the students and whether the concept can also be applied to several hours with different content. In this context, the reduced lower variance of the treatment group results in the algorithmic thinking test noted above can also be investigated more closely, possibly with the potential for improvement of weaker students about algorithmic thinking.

Even though, this pilot study was carried out completely online due to the COVID-19 pandemic, future studies will be carried out in person but we are considering whether the worksheets will continue to be made available online or whether it will be easier for the students to handle in paper form. It is also worth considering integrating notional machines into the students' problem-solving steps.

Furthermore, as next steps the SWAT concept will be applied in combination with different educational digital devices as for instance robots. Such variation of methodology could determine to what extent the concept can be implemented and the measuring instruments can be retained or must be adapted.

References

Barr, V., Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54. `https://doi.org/10.1145/1929887.1929905`

Cansu, F.K., Cansu, S.K. (2019). An Overview of Computational Thinking. *International Journal of Computer Science Education in Schools*, 3(1), 17–30. `https://doi.org/10.21585/ijcses.v3i1.53`

Csizmadia, A., Standl, B., Waite, J. (2019). Integrating the constructionist learning theory with computational thinking classroom activities. *Informatics in Education*, 18(1), 41–67.

Curzon, P., Waite, J., Maton, K., Donohue, J. (2020). Using semantic waves to analyse the effectiveness of unplugged computing activities. *ACM International Conference Proceeding Series*. 9781450387590. `https://doi.org/10.1145/3421590.3421606`

Eickelmann, B., Bos, W., Gerick, J., Goldhammer, F., Schaumburg, H., Schwippert, K., Senkbeil, M., Vahrenhold, J. (2019). *ICILS 2018 Deutschland – Computer-und informationsbezogene Kompetenzen von Schülerinnen und Schülern im zweiten internationalen Vergleich und Kompetenzen im Bereich Computational Thinking*. Waxmann Verlag, Münster. 978-3-830-99000-0.

Fincher, S., Jeuring, J., Miller, C.S., Donaldson, P., Du Boulay, B., Hauswirth, M., Hellas, A., Hermans, F., Lewis, C., Mühling, A., Pearce, J.L., Petersen, A. (2020). Notional Machines in Computing Education: The Education of Attention. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, 21–50. 9781450382939. `https://doi.org/10.1145/3437800.3439202`

Fraillon, J., Ainley, J., Schulz, W., Duckworth, D., Friedman, T. (2019). IEA International Computer and Information Literacy Study 2018 Assessment Framework. *IEA International Computer and Information Literacy Study 2018 Assessment Framework*, 25–31. 9783030193898.

https://doi.org/10.1007/978-3-030-19389-8

Hoijtink, H., Mulder, J., van Lissa, C., Gu, X. (2019). A Tutorial on Testing Hypotheses Using the Bayes Factor. *Psychological Methods*. https://doi.org/10.1037/met0000201

Klepsch, M., Schmitz, F., Seufert, T. (2017). Development and validation of two instruments measuring intrinsic, extraneous, and germane cognitive load. *Frontiers in Psychology*, 8(NOV). https://doi.org/10.3389/fpsyg.2017.01997

Maton, K. (2013). Making semantic waves: A key to cumulative knowledge-building. *Linguistics and Education*, 24(1), 8–22. https://doi.org/10.1016/j.linged.2012.11.005 http://dx.doi.org/10.1016/j.linged.2012.11.005

Mayring, P., Fenzl, T. (2019). *Handbuch Methoden der empirischen Sozialforschung*. Springer Fachmedien Wiesbaden, ???, pp. 633–648. 9783658213084. https://doi.org/10.1007/ 978-3-658-21308-4. http://dx.doi.org/10.1007/978-3-658-21308-442

Moreno-León, J., Robles, G. (2015). Dr. Scratch: A web tool to automatically evaluate scratch projects. *ACM International Conference Proceeding Series*, 09-11-Nove(October), 132–133. 9781450337533. https://doi.org/10.1145/2818314.2818338

Paas, F.G., Van Merriënboer, J.J., Adam, J.J. (1994). Measurement of cognitive load in instructional research. *Perceptual and Motor Skills*, 79(1 Pt 2), 419–430. https://doi.org/10.2466/pms.1994.79.1.419

Papavlasopoulou, S., Giannakos, M.N., Jaccheri, L. (2019). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior*, 99, 415–427.

Papert, S. (1982). *Mindstorms*. Birkhäuser Basel, ??? https://doi.org/10.1007/ 978-3-0348-5357-6.

Park, Y., Shin, Y. (2019). Comparing the effectiveness of scratch and app inventor with regard to learning computational thinking concepts. *Electronics (Switzerland)*, 8(11). https://doi.org/10.3390/electronics8111269

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67. https://doi.org/10.1145/1592761.1592779

Robert K. Yin (2018). *Case Study Research and Applications Design and Methods Sixth Edition*. https://www.amazon.co.jp/dp/B07C66LQX2/ref=dp-kindle-redirect?encoding = *UTF8btkr = 1StudyResearchandApplications − DesignandMethods?text = CaseStudyResearchandApplications − DesignandMethodsgroup$_h$old*

Román-González, M. (2015). Computational Thinking Test: Design Guidelines and Content Validation. *Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015)*, (July 2015), 2436–2444. 9788460682431. https://doi.org/10.13140/RG.2.1. 4203.4329

Sadykova, O.V., Il'bahtin, G.G. (2020). The Definition of Algorithmic Thinking, 113(Fred 2019), 419–422. https://doi.org/10.2991/fred-19.2020.85

Schwarz, R., Hellmig, L., Friedrich, S. (2021). Informatikunterricht in Deutschland – eine übersicht. *Informatik-Spektrum*, 44(2), 95–103. https://doi.org/10.1007/s00287-021-01349-9

Skulmowski, A., Rey, G.D. (2020). Subjective cognitive load surveys lead to divergent results for interactive learning media. *Human Behavior and Emerging Technologies*, 2(2), 149–157. https://doi.org/10.1002/hbe2.184

Smith, S.W., Daunic, A.P., Taylor, G.G. (2007). Treatment fidelity in applied educational research: Expanding the adoption and application of measures to ensure evidence-based practice. *Education and Treatment of Children*, 30(4), 121–134. https://doi.org/10.1353/etc.2007.0033

Standl, B. (2017). Solving everyday challenges in a computational way of thinking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10696LNCS(November),180–191.9783319714820.https://doi.org/10.1007/978-3-319-71483-7_15

Tissenbaum, M., Ottenbreit-Leftwich, A. (2020). A vision of K—. *Communications of the ACM*, 63(5), 42–44. https://doi.org/10.1145/3386910

Waite, J., Maton, K., Curzon, P., Tuttiett, L. (2019). Unplugged computing and semantic waves: Analysing crazy characters. *ACM International Conference Proceeding Series*. 1595930361. https://doi.org/10.1145/3351287.3351291

Wing, J.M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

F. Ritter is pre-doc research assistant at the Department for Informatics and Digital Education at the Karlsruhe University of Education in Germany. Her research interests are in the development of teaching/learning concepts for lower secondary for promoting secondary school students' algorithmic thinking competencies integrating block-based programming languages, robots, and on in the training of pre-service Informatics teachers in the Teaching-Learning-Lab for Informatics at the same place.

B. Standl is Professor for Computer Science Education at the Karlsruhe University of Education in Germany. His research interests are in the conceptualisation of teaching-learning scenarios and in problem-solving strategies based on computational thinking concepts.