Using Computational Thinking to Facilitate Language Learning: A Survey of Students' Strategy Use in Austrian Secondary Schools

Marina Rottenhofer
Johannes Kepler University Linz, Austria

Lisa Kuka
Johannes Kepler University Linz, Austria

Sandra Leitner
Johannes Kepler University Linz, Austria

Barbara Sabitzer
Johannes Kepler University Linz, Austria

Abstract

After Jeanette Wing in 2006 described computational thinking (CT) as a fundamental skill for everyone just like reading or arithmetic, it has become a widely discussed topic all over the world. Computational thinking is a problem-solving skill set that is used to tackle problems in computer science. However, these skills, such as pattern recognition, decomposition, abstraction, generalization, and algorithmic thinking, are useful in other domains, as well. This study focuses on the use of CT skills to approach complex linguistic learning tasks in the foreign language classroom. To foster these problem-solving skills, an innovative method is used. The authors take advantage of computer science (CS) models (e.g. Unified Modeling Language – UML) and transform them into a teaching and learning tool. This paper describes the design and implementation of a survey used to detect students' use of learning strategies that are linked to computational thinking. This survey is an instrument used in a multiple-case study and was administered at the beginning of the interventions. The participants of the study were learners of English and Spanish (n=66) from two secondary schools. Results indicated that the students were medium to low users of learning strategies that demand problem-solving skills related to computational thinking. Differences by gender were also found, with females reporting higher use of learning strategies than males. To conclude, the study showed a low use of strategies among students and highlighted the importance of introducing students to learning strategies and fostering skills needed for future professional life.

Keywords: computational thinking, digital literacy, foreign language learning, learning strategy, modeling, visualization

Fast technological development shapes our future and has an impact on our personal, social as well as professional lives. For this reason, schools are confronted with high demands to equip students with knowledge and skills that help them to cope with the challenges of the future. According to the Future of Jobs Report 2020 (World Economic Forum, 2020), the top skills required in 2025 are divided into four groups: problem-solving, self-management, working with people, and technology use and development. Analytical thinking, active learning, and learning strategies as well as complex problem-solving are at the very top of this ranking. One problem-solving skill set, which has the potential to prepare students for future demands is computational thinking (CT).

Since 2006, CT has gained considerable attention as one of the core skills next to reading, writing, as well as arithmetic (Wing, 2006) and has already become part of compulsory education in many countries, including Austria (BMBWF, 2018). With this transformation, CT and CS models have found their way into the foreign language classroom as well. In our multiple case study that is based on Yin's model (2009), diagrams from the field of computer science (CS) are implemented as a teaching and learning strategy to foster computational thinking in foreign language education. In computer science, on the other hand, diagrams based on the UML (Unified Modeling Language) [Seidl et al., 2015] or Chen notation (Chen, 1976) are used to visually depict software systems or database structures. With the use of these diagrams in a different context as a teaching and learning strategy, the authors reach several goals at once. Firstly, many years of implementation and research have shown that modeling with CS diagrams is a useful visualization strategy for learners of all ages, is easy to acquire for teachers and students, and is applicable in all subjects (Demarle-Meusel et al., 2020; Rottenhofer et al., 2021; Sabitzer & Pasterk, 2015). Secondly, learners get in contact with a repertoire of static and dynamic CS diagrams outside computer science lessons which may help them to familiarize themselves with this field, spark their interest, and introduce basic computer programming concepts. Thirdly, depicting learning content with a model requires cognitive flexibility and fosters computational thinking skills such as abstraction, generalization, pattern recognition, and algorithmic thinking. To summarize, learners do not only get in touch with computer science concepts but also receive a useful learning tool that they can apply in different learning settings to solve complex tasks and memorize information long term. In the current research, CS models are implemented as graphic organizers in several foreign language learning settings. This paper presents the results of a survey that learners received at the beginning of the intervention. This survey aimed to examine to what extent the participants use learning strategies that are connected to computational thinking. For this, a survey on learning strategies had been modified from the two German questionnaires LSN – *Learning Strategy Use* (Martin & Nicolaisen, 2015) and LIST – *Learning Strategies at University* (Wild & Schiefele, 1994) by linking it to the areas of computational thinking.

Literature Review

In the 1980s, computational thinking (CT) was first mentioned by Papert (1980) in his work on teaching computer literacy at an early age where he saw CT as the result of his constructionist learning theory. Twenty-six years later, the term was boosted by Jeanette Wing as "a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" (2006, p. 1). Since then, much research has been done and numerous definitions emerged, many of which focus on programming, leading to the assumption that programming is a necessary tool to teach CT (Voogt et al., 2015). However, everyone should acquire CT, not only programmers (National Research Council, 2010) and students should get exposed to CT long before programming (Lu & Fletcher, 2009). To date, several researchers have investigated the integration of CT in foreign language learning (FLL) [Barr & Stephenson, 2011; Hsu & Liang, 2021; Lu & Fletcher, 2009; Parsazadeh et al., 2021]. However, to the best of the authors' knowledge, none of them have investigated hands-on approaches to foster CT in FLL in depth.

In this study, computer science (CS) models are used as a form of graphic organizer (GO) to foster CT skills and get students engaged with computer science concepts outside the CS lessons. GOs originally derive from Ausubel's cognitive learning theory (1962), where he applied them as advance organizers at the beginning of the learning process. A graphic organizer is defined as a "visual and graphic display that depicts the relationships between facts, terms, and or ideas within a learning task" (Hall & Strangman, 2002, p. 2). According to Willis (2007, p. 315), this creative approach "coincides with the brain's style of patterning" and allows students to connect the information to previously stored memories, cluster information, discover patterns, and sort and store new data. This description is well-aligned with CT and demonstrates the usefulness of using models to foster these problem-solving skills. Furthermore, according to research, the use of GOs is particularly useful for students with learning difficulties (Dexter & Hughes, 2011; Kim et al., 2004; Sousa, 2017). These results confirm the authors' experiences of the benefit of modeling, especially for pupils with learning deficits. A major cause of learning difficulties in FLL such as dyslexia lies in struggles with recognizing and using language patterns in the new language. Even if pupils suffer from dyslexia, they may still have good intellectual abilities. However, they may not be able to notice similarities and differences between vocabulary and word formation patterns (i.e. semantic processing) in the foreign language compared to their native language (Schneider & Crombie, 2012).

The difficulties in recognizing language patterns make learning difficult. However, modeling with CS diagrams can support these pupils in their learning process. By teaching with appropriate diagrams in common FLL environments, all pupils, but especially pupils with learning difficulties, benefit as they acquire learning content easier and thereby learn to speak the foreign language more effectively. The following sub-section presents learning theories connected to graphic organizers and computational thinking.

Modeling, Computational Thinking, and Theories of Learning

The use of CS models as GOs is a teaching method that combines cognitivist and constructivist learning theories and computer science concepts to foster computational thinking skills.

Cognitivism emerged in the late 1950s and, in comparison to behaviorism that is based on the stimulus-response theory, relied on cognitive sciences by focusing on cognitive processes (Ertmer & Newby, 2013). Several cognitive learning theories support the use of GOs such as the subsumption theory, schema theory, dual coding theory, and cognitive load theory. According to Ausubel's (1962) subsumption theory on meaningful learning, learning and retention are facilitated when new information is related to already existing cognitive structures. To achieve this, he suggested the use of advance organizers. Anderson and Pearson (1988) claimed that the subsumption theory is consistent with his schema theory, where a person has understood a text when they have found a mental "home" for the information in the text, or else "that he or she has modified an existing mental home in order to accommodate that new information" (Anderson & Pearson, 1988, p. 2). The dual coding theory postulates that there are two systems, verbal and imagery, for processing information (Clark & Paivio, 1991). In other words, when information is presented in both forms, e.g. verbally and visually with a model, chances of retrieval are increased. Lastly, the cognitive load theory by Sweller et al. (1998) assumes that the working memory has a limited capacity and can therefore only deal with a limited amount of information at a certain time. Used appropriately, GOs can reduce cognitive load and lead to better learning outcomes (Rahmat, 2020).

Constructivism is often considered a branch of cognitivism. However, the main difference is that constructivist psychologists believe "that the mind filters input from the world to produce its own unique reality" (Ertmer & Newby, 2013, p. 55). In other words, what we know of the real world is constructed personally with our own interpretations—"humans create meaning as opposed to acquiring it" (Ertmer & Newby, 2013, p. 55). Out of Piaget's constructivism, Papert developed the learning theory constructionism, where the focus shifts "from universals to individual learners' conversation with their own favorite representations, artifacts, or objects-to-think with" (Ackermann, 2001, p. 4). According to Ali and Yahaya's systematic review, constructivist learning theory is primarily used in computational thinking focusing on primary and secondary school levels, followed by constructionism (2020). However, they also claim that there are many studies on CT that do not focus on learning theories at all. Bellettini et al. postulate a social-constructivism approach to informatics and CT where the teacher's role is to "support the construction of knowledge through setting up contexts and scaffolding material favoring the activation of the learning process, in which the ultimate actor is the learner itself" (2018, p. 4). This means that teachers should motivate students to use active techniques in their learning process.

Computational Thinking and Language Learning

This section describes the core elements of computational thinking that are the focus of the current study. In the literature, CT is represented with different manifestations, core concepts, and skills. The Joint Research Center (JRC) from the European Commission (Bocconi et al., 2016) conducted a literature review and analyzed the skills emerging from the most prominent papers on CT. As a result, they developed a list of core elements, which are abstraction, algorithmic thinking, automation, decomposition, debugging, and generalization. In this study, the authors refer to the elements proposed by the JRC, extend them with pattern recognition (Curzon et al., 2019), and link them to foreign language teaching. Additionally, this section gives best practice examples on how to use modeling and CT as techniques that support students in creating new knowledge and engaging them actively in the learning process.

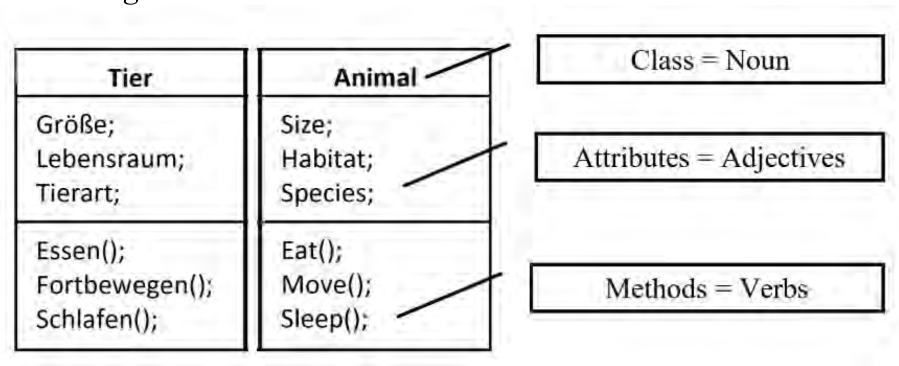
Decomposition

Decomposition is the process of dividing a bigger problem into smaller sub-problems (Barr & Stephenson, 2011). This divide-and-conquer strategy helps to facilitate the understanding of a problem and, thus, can be solved systematically as well as individually. In language education, this is a skill widely used. For example, when writing a paper only a few people would write it straight from the beginning to the end. Usually, the structure of it is well thought-through and headlines like "introduction", "methodology", "conclusion", and so forth. are created first. Then, additional arguments or topics are found for the main body. The introduction and conclusion are also known to be written last. This process illustrates decomposition at its best.

Abstraction

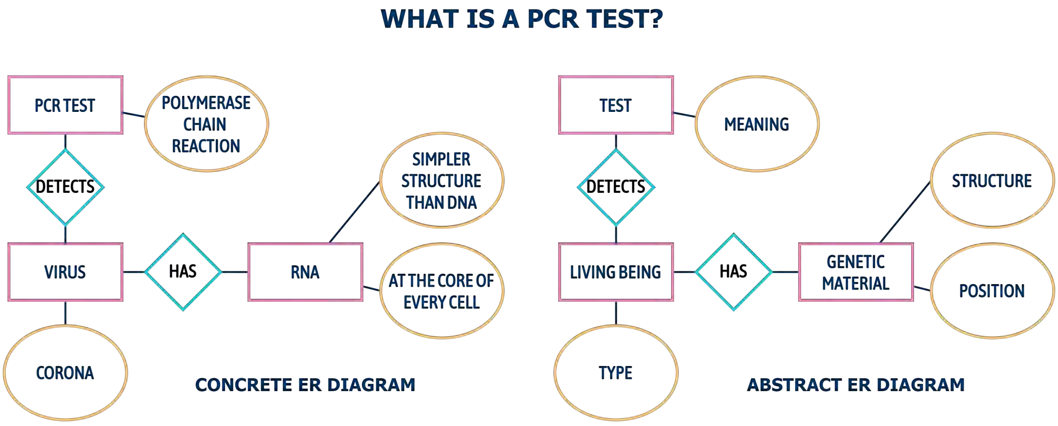
Abstraction describes the process of reducing complexity by omitting unnecessary details. Thus, the main characteristics of a problem or item are defined. Everyone handles abstract objects daily, for example, when using a map. Every map is a simplified presentation of reality. When learning about giving directions in the language classroom, subway maps are a common tool taken from real life. Another example is writing a summary. A summary is characterized by leaving out unnecessary details and concentrating on the most important information. Hence, training on writing summaries and encouraging students to take notes or highlight important information in a text, also helps to strengthen computational thinking skills. In computer science, class and object diagrams are used to visualize various components of a system and their relations (Seidl et al., 2015). Whereas class diagrams describe the abstract model of a system (e.g. animal), object diagrams illustrate concrete objects (e.g. cats and dogs). In the language classroom, these models can be used to develop new vocabulary about specific topics, illustrate relations and hierarchies, and categorize these items. Figure 1 shows a simple example of one class. As can be seen, the name of a class is always a noun, attributes are seen as adjectives, and methods as verbs. Thus, students can also practice the difference between these word classes and word formation.

Figure 1
Class Diagram



Another model, which is used in computer science frequently, is the entity-relationship model (ER model) [Chen, 1976]. It consists of three elements – rectangles as "entity-types" that are used as nouns, diamond shapes as "relationship types", and the ellipses as "attributes" that describe the characteristics of the nouns. The ER model can be used as an intermediate step when writing summaries, supporting especially students with learning difficulties when writing texts. Figure 2 shows a model where elements of a text on COVID-19 were transformed into an ER diagram with concrete and generalized terms. Usually, in computer science, the ER diagram only uses generic terms instead of specific terms since it represents a type of a system and not an instance (Bagui & Earp, 2003). However, in the language classroom, this can be adapted by using concrete terms of a text and/or abstract terms.

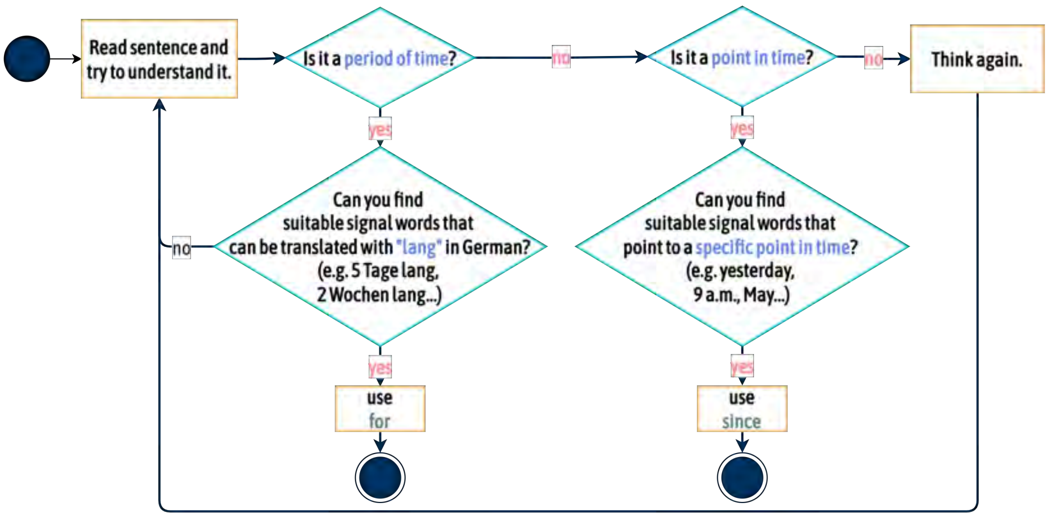
Figure 2
Entity-Relationship Diagram



Pattern Recognition and Generalization

Finding patterns is something inherently human, and the brain can remember patterns more easily (Grabmeier, 2018). As soon as patterns, similarities, and connections are found, a generalization of these can be done, and already known problem-solving strategies which worked for a similar scenario can be re-used. Also, in many cases, it is possible to draw conclusions from a part or general to the whole. Every language educator who used an inductive method is already familiar with pattern recognition and generalization. For example, the teacher provides various grammatical items such as sentences in the past tense using regular verbs. Subsequently, the students have to find grammatical rules based on the examples given. Figure 3 illustrates how the use of an activity diagram can visualize the grammatical rules, such as the use of “for” and “since” in English. Also, it can function as a step-by-step guide.

Figure 3
Activity Diagram showing the Use of For and Since in English



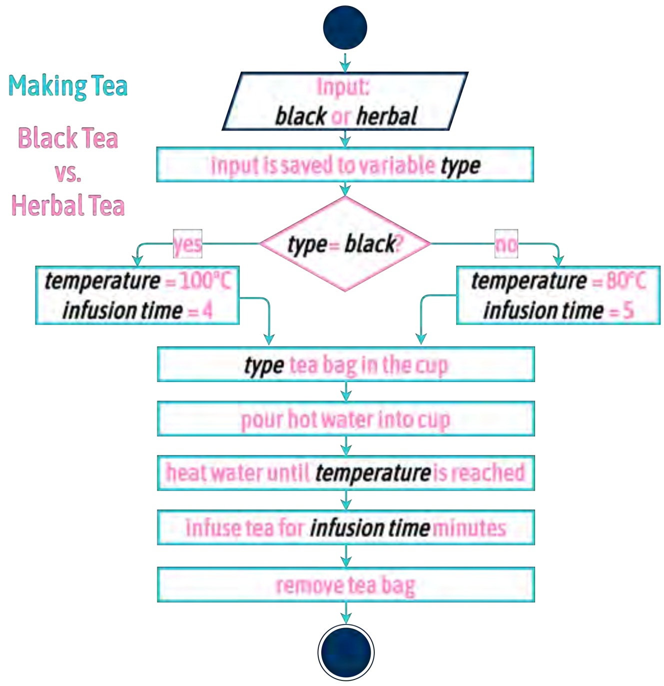
Another example in which generalization in the language classroom is used is by giving examples and prompts in which generalized terms like genre, title, author, and so on, are used.

The students then have to find the actual genre, title, and author of the presented text, that is, gothic novel, Frankenstein, and Mary Shelley.

Algorithmic Thinking

An algorithm is often described as a step-by-step guide comparable to a recipe. Teaching students to write good recipes can be compared to writing an algorithm. Not only is it important to be precise in its formulation, but also to think systematically about which step comes after the other. How long do you have to beat the eggs to make your cake heavenly fluffy? Usually, teachers give the exercise to simply write a recipe, but for students with learning difficulties, it may be a good idea to sketch the information at first via an activity diagram. With this intermediate step, they not only have the structure first but also the key vocabulary needed for the exercise. Figure 4 shows an example of an activity diagram created for a recipe.

Figure 4
Algorithm for Making Tea



Testing and Debugging

It is not enough to find solutions for problems; it is also necessary to systematically analyze these solutions using skills such as testing, tracing, as well as reasoning. Based on this accurate analysis, errors can be fixed and results predicted and verified. In the language classroom, students can be trained to achieve this by correcting (one's own) errors, for example, in a filling the gap exercise or when learning how to give feedback.

Automation

Automation is a work-saving process in which a machine or computer is instructed to perform a series of repetitive tasks quickly and efficiently compared to the processing power of a human. This is the only skill that usually is not very common in the language classroom, although there would be possibilities to include programming as well, for example, with the use of the programming language Scratch or exercises from machinelearningforkids.com.

Methods

Background

In the school year 2020/21, a multiple case study (Yin, 2009) on modeling as a teaching and learning strategy to foster computational thinking was conducted. The subjects of the case studies were partner schools of the COOL (computer sciences-supported, cross-curricular, and cooperative open learning) Lab at the Johannes Kepler University in Linz. The JKU COOL Lab is an innovative teaching and learning lab for teachers, children of all ages, and university students. It focuses on computer science, computational thinking, and digital literacy. The lab has many offerings including workshops, weekly clubs for gifted students, theater shows on digital education, teacher training, and so forth. In addition to offerings for all interested parties, the lab works intensively with several partner schools where projects are implemented and researched over a longer period. In the multiple case study, modeling was implemented in four foreign language classes of two partner schools to find out more about (1) teachers' and students' perceptions of modeling as a teaching and learning strategy, (2) the chances and challenges of the implementation of modeling and (3) computational thinking as a problem-solving strategy. This paper focuses on computational thinking as a problem-solving strategy and presents the results of a survey administered to all the participants of the multiple case study at the beginning of each of the interventions. This survey aimed to find out more about students' use of learning strategies that are related to computational thinking. In particular, the following research questions were explored:

1. Is there a connection between learning strategies and the areas of computational thinking as a problem-solving strategy?
    a. If yes, what strategies are associated with computational thinking?
2. Do students use strategies associated with computational thinking to better understand and process learning content?
3. Does the use of learning strategies differ by gender?

Participants

The questionnaire was administered to a total of 66 students ($n_f = 31$, $n_m = 35$) from two partner schools (PS_n) of the JKU COOL Lab. In those partner schools, several teachers collaborated intensively with the researchers and two of them were willing to participate in this study. Thus, random sampling was not possible. Before conducting the study, written permission was obtained from the school principals as well as the parents of the participants. Both groups of PS1 (English class) and PS2 (Spanish class) were involved in the multiple case study for several months working with models as a teaching and learning strategy to foster computational thinking skills. To get an insight into students' computational thinking strategy use, the survey was administered at the beginning of the intervention. In the English group composed of 51 students, there were 29 males and 22 females with a mean age of 14.25 and a standard derivation of 1.369. The Spanish group consisted of 15 students, 6 males, and 9 females with a mean age of 13.27 and a standard deviation of 1.981. At the beginning of the study, none of the students were familiar with modeling and the concept of computational thinking. The demographic information of the participants is presented in Table 1.

Table 1
Participants in the Study

School	Subject	N	Male	Female	Mean Age	SD
PS1	English	51	29	22	14.25	1.369
PS2	Spanish	15	6	9	13.27	1.981

Instrument

In this study, a paper-based questionnaire on learning strategies was administered, consisting of 37 Likert-formatted items. For this survey, the authors adopted items from the LSN (*Learning Strategy Use*) questionnaire from Martin and Nicolaisen (2015) and combined it with four items from the LIST (*Learning Strategies at University*) questionnaire (Wild & Schiefele, 1994) bringing it up to 37 items.

The four LIST items were the following:

1. I try to organize the material so that I can easily remember it.
2. I visualize the material to be learned.
3. I learn key terms by heart to help me remember important areas of content.
4. I memorize a self-made overview with the most important terms.

The frequency was measured with a five-point Likert scale ranging from 1 (very rarely) to 5 (very often). The questionnaire was issued in German and was translated for this paper.

Procedure

The questionnaire was administered to the students at the beginning of the multiple case study in their regular language lessons. The participants of the study had no previous knowledge of modeling and computational thinking. The survey had no time limit to make sure the students were not under any pressure and could think deeply about their answers. The students needed approximately 10-15 minutes to respond to all the items of both Part 1 and Part 2 of the questionnaire.

To identify which learning strategies are used that relate to CT skills and visualization, three experts independently analyzed the first part of the questionnaire and filtered out the items (1-37) that can be assigned to the CT skills mentioned in section 2 on the one hand and to visualization strategies on the other. After this analysis, the experts discussed the respective selection and decided on the items used and their assignment to the respective categories. The statistical analysis was then conducted using the statistical software IBM SPSS Statistics 23.

Findings

Learning Strategies Related to Computational Thinking

The first research question sought to answer whether there is a connection between learning strategies and the core elements of computational thinking and if yes, which ones. The expert analysis has shown that a total of 22 Likert items can be related to computational thinking and visualization strategies. Specifically, 18 items from the Martin and Nicolaisen questionnaire

(2015) relate to computational thinking, and two items each of the LIST relate to visualization and CT (Wild & Schiefele, 1994). Table 2 shows an overview of the remaining items and the allocation to the individual areas. Since all the CT skills are intertwined, some items have multiple assignments. From the core CT skills proposed in section 2, all the skills except “automation” could be associated with items in the questionnaire.

Use of Strategies Related to Computational Thinking

The second research question investigated whether students use strategies associated with computational thinking to better understand and process learning content. To find out which of the CT skills according to the learning strategies are used the most, descriptive statistics, including means and standard deviation of the six CT categories as well as the category related to visualization was used. As illustrated in Table 3, three categories are above the middle of the Likert scale and four are below it. Testing and debugging strategies are used most frequently (M= 3.28; SD= .76), closely followed by decomposition strategies (M= 3.10; SD= .78) and algorithmic thinking (M= 3.03; SD= .79). On average, the categories below the mid point are: generalization (M= 2.88; SD= .90), abstraction (M= 2.83; SD=.89), pattern recognition (M=2.78; SD= .87) and lastly, visualization (M= 2.39, SD=1.09). According to Table 3, all categories had a mean score at the medium or low level. None of the categories had a mean value at a high level above 4.0.

Besides the descriptive analysis of the seven categories mentioned above, individual items were also ranked and highlighted as the five most and least commonly used learning strategies. As seen in Table 4, the most common strategy is to use the internet or dictionary when words are unclear (M=3.91; SD=1.32), whereas the least common strategy (see Table 5) is to create drawings or sketches to better see how things belong together (M=1.95; SD=1.07). Looking at all 22 items, none of the items has a mean value at a high level above 4.0. Half of the items (N=11) have a mean score at the medium level above 3.0, whereas 10 items are above 2.0 and only one item below.

Table 2
Survey Part 1: Likert-Items

Nr.	Item	Category
1	When I have to study for an exam, I make a short summary.	AB
2	I often do drawings or sketches to better see how things belong together.	V
3	I underline the important passages in the textbook.	AB
4	I try to organize the material so that I can easily remember it.	AL
5	I visualize the material to be learned.	V
6	When I learn something new, I try to figure out what to do with that knowledge (what is the practical use?).	GE
7	I wonder how what I am learning relates to what I have known so far.	AB
8	I wonder if what I am learning or hearing is logical.	TD
9	I wonder if there could be other explanations for what I read or hear.	GE, TD, PR
10	Instead of studying for a long time, I spread the work over several days.	DC
11	I repeat things (such as foreign language vocabulary) in small portions, but regularly (e.g. every day for 10-15 min).	DC, AL, PR
12	I learn key terms by heart to help me remember important areas of content.	AB, GE
13	I memorize a self-made overview with the most important terms.	AB, GE
14	When my learning is not going well, I try to change something and see if it goes better.	TD
15	Before I start to work, I set myself clear goals.	DC
16	While studying, I check whether I am still on the right track.	TD
17	When I stop working, I check whether I have achieved my goals.	TD
18	When I study, I make a realistic schedule.	AL, DC
19	I make sure that I have enough time the day before an exam to review all of the material again.	DC
20	Before an exam or a lecture, I think about what to do if things do not go well.	AL
21	I look for more information in books or on the Internet if something is not quite clear to me.	TD
22	If I do not understand words, I look them up on the Internet or in a dictionary.	TD

Abbreviations: DC Decomposition, PR Pattern Recognition, AB Abstraction, AL Algorithmic Thinking, GE Generalization, TD Testing & Debugging, V Visualization

Table 3
Descriptive Statistics – Computational Thinking Skills. N= 66

	Minimum	Maximum	Mean	Std. Deviation
Decomposition	1.40	5.00	3.1010	.77854
Pattern Recognition	1.00	4.67	2.7778	.86791
Abstraction	1.00	4.75	2.8258	.89224
Algorithmic Thinking	1.25	5.00	3.0253	.79124
Generalization	1.00	4.50	2.8750	.90219
Testing & Debugging	1.29	4.86	3.2835	.75696
Visualization	1.00	4.50	2.3939	1.09374

Table 4
Top 5 of the Most Commonly Used Learning Strategies. N= 66

	Nr.	Category	Min.	Max.	Mean	Std. Dev.
1. If I do not understand words, I look them up on the Internet or in a dictionary.	22	TD	1	5	3.91	1.321
2. I wonder if what I am learning or hearing is logical.	8	TD	1	5	3.73	1.089
3. I make sure that I have enough time the day before an exam to review all of the material again.	19	AL	1	5	3.46	1.251
4. I try to organize the material so that I can easily remember it.	4	AL	1	5	3.41	1.265
5. Before I start to work, I set myself clear goals.	15	DC	1	5	3.38	1.034

Abbreviations: DC Decomposition, AL Algorithmic Thinking, TD Testing & Debugging

Table 5
Top 5 of the Least Commonly Used Learning Strategies. N= 66

	Nr.	Category	Min.	Max.	Mean	Std. Dev.
1. I often do drawings or sketches to better see how things belong together.	2	V	1	5	1.95	1.073
2. When I have to study for an exam, I make a short summary.	1	AB	1	5	2.52	1.099
3. When I study, I make a realistic schedule.	18	AL, DC	1	5	2.61	1.341
4. I wonder how what I am learning relates to what I have known so far.	7	AB	1	5	2.65	1.295
5. While studying, I check whether I am still on the right track.	16	TD	1	5	2.73	1.103

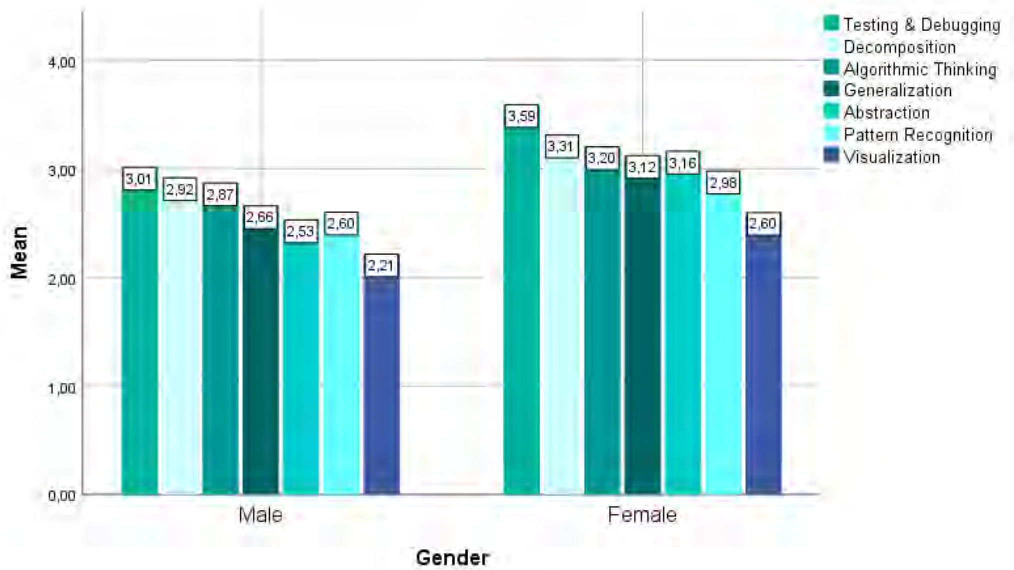
Abbreviations: DC Decomposition, AB Abstraction, AL Algorithmic Thinking, TD Testing & Debugging, V Visualization

Learning Strategies, Computational Thinking, and Gender

The last research question sought to answer whether strategy use related to CT differs by gender. The independent Sample T-Test revealed that female students reported statistically more frequent use of learning strategies related to CT than male students did. Female students have a higher mean score in relation to all learning strategies (M_f= 3.31; SD_f= .58, M_m= 2.80; SD_m= .63, p<.05) as well as in the different CT categories (see Figure 5). However, when looking at the single CT categories, only decomposition, abstraction, generalization, and testing and debugging were found to be statistically different (P<.05).

Figure 5

Mean Score of CT Strategy Use Related to Gender. N=66



A Pearson correlation coefficient was computed to assess the linear relationship between gender and CT categories, as well as overall strategy use. As reported in Table 6, there is a statistically positive correlation between gender (1=male, 2=female) and decomposition, abstraction, generalization, and testing and debugging as well as the overall strategy use.

Table 6

Pearson Correlation Coefficient between Gender and Strategy Use. N= 66

		gender	DC	PR	AB	AT	GE	TD	V	SUM
gender	Pearson Correlation	1	.254*	.219	.357**	.211	.259*	.384**	.176	.393**
	Sig. (2-tailed)		.039	.077	.003	.088	.036	.001	.158	.001
	N	66	66	66	66	66	66	66	66	66

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Discussion

This research sought to investigate the connection between learning strategies and the areas of computational thinking as well as students' use of the respective strategies. The results of the survey indicate that the participants were medium to low users of learning strategies that demand problem-solving skills related to computational thinking. These findings are consistent with previous studies on learning strategy use. For example, Aslan (2009) investigated language learning strategies and also found a medium level of strategy use regardless of gender. However, he found that higher-achieving students use more learning strategies. The low use of learning strategies, in general, may have several reasons. One explanation may be that students do not know which strategies are effective (Morehead et al., 2016). Korenell and Bjork (2007), on the other hand, found that many students' goal is to pass exams and not to store information long-term. Another possible explanation for the rare use of learning strategies could be the time factor. Previous studies have shown that many students do not split learning content over a longer period, but rather wait until just before an exam, often until the last day (Blasiman et al., 2017; Susser & McCabe, 2013; Taraban et al., 1999). There are similarities between the attitudes expressed by the participants of this study and those mentioned above. It is apparent from Tables 4 and 5 that many students make sure to have enough time the day before the exam (Item 19) and pay less attention to making a realistic schedule (Item 18) or checking whether they are still on the right track (Item 16). Hence, it could be hypothesized that the lack of time is the reason why students prefer quick searches for information (Item 22) rather than time-consuming strategies (Items 1, 2). Time constraints could also be the reason why students are less concerned about linking new information to prior knowledge (Item 7). However, it seems that students still organize their work, try to set goals (Items 4, 15), and question the new information (Item 8).

The results also demonstrate a statistically significant difference in learning strategy use by gender and correlate well with previous studies in the context of language learning where females surpassed males. In his work on language learning, Oxford reports on females "using more varied strategy types and employing strategies more frequently than males" (1993, p. 85). Furthermore, he claims that when students are not explicitly asked by the teacher to use a certain L2 learning strategy, they tend to use those favoring their learning style. For example, analytic learners (often males) prefer strategies involving logic, whereas the global learner (often females) prefer to use social strategies including searching for the main idea and intuitively guessing. In a study on gender and language learning strategies in learning English, Aslan (2009) also found a significant difference in strategy use, indicating that females, on average, employed more strategies than males in all domains and subscales investigated.

Although a great amount of literature reports a significant gender difference proposing that females generally use more learning strategies than males, few studies came to the opposite conclusion. For instance, Tercanlioglu (2004) conducted a study on foreign language learning strategies with 184 pre-service teachers from Turkey, showing a gender difference favoring males. According to her, the cultural background could be one of the reasons that the results are not consistent with many previous studies.

Limitations

This survey helped to illuminate strategy use of students and served as the basis for the implementation of CS modeling in foreign language learning to foster computational thinking skills. Nevertheless, the study also has its limitations. One of them includes the self-selection bias resulting from the collaboration with the partner schools of the JKU COOL Lab. Another limitation of this study is the sample size. Further research and wider trials are needed to be able to generalize the results and to determine which other factors besides gender influence strategy use. Moreover, to be able to fully understand this phenomenon, the use of further data-gathering instruments such as interviews is also advisable, so that the case can be viewed from different angles leading to richer results and conclusions. A major reason why only the questionnaire was used at the time of the study was due to the difficult circumstances caused by the COVID-19 pandemic. Therefore, further investigations with interviews are planned to get a more holistic picture.

Recommendations and Conclusion

This survey aimed to investigate the use of learning strategies that can be linked to the core elements of computational thinking (CT). For this, an expert group analyzed and identified items of the two German questionnaires LSN (*Learning Strategy Use*) [Martin & Nicolaisen, 2015] and LIST (*Learning Strategies at University*) [Wild & Schiefele, 1994], and developed a list of learning strategies related to computational thinking. By analyzing the degree of strategy use among students, this study established that all participants in the study were only medium to low degree strategy users. Furthermore, results show that females reported statistically higher use of learning strategies related to CT than male students. When looking at the six CT skills as well as visualization strategies, testing and debugging strategies marked the highest usage, closely followed by decomposition strategies and algorithmic thinking. The category of visualization skills occupied the last place in the ranking. Concerning individual strategies, item 22 (If I do not understand words, I look them up on the Internet or in a dictionary) was the most frequently used strategy, and item 2 (I often do drawings or sketches to better see how things belong together) was the least frequently used strategy.

These results indicate that although students are generally medium to low users of strategies, they prefer fast strategies like researching information online to techniques that are more time-consuming, such as visualization strategies. It is also possible that students are not aware of the effectiveness of various strategies, especially for retaining information long-term. CT skills such as decomposition, abstraction, pattern recognition, and algorithmic thinking are essential for future professional life. Thus, an important implication is that teachers should raise strategy awareness and offer students opportunities to gain these skills by providing suitable activities such as modeling. With this approach, students' interest in more time-consuming visualization strategies can be increased as they might see long-term benefits that outweigh expenditure of time.

The results of this survey work as the basis for the implementation of computer science models as a teaching and learning strategy to foster CT skills. The experience and research on modeling and CT in language teaching and other subjects have shown promising results in recent years. Nevertheless, future work is planned to investigate the reasons behind the low use of strategies generally and visualization techniques in particular. Moreover, further studies could shed more light on the contribution of higher CT strategy use on learning achievement. To conclude, with modeling as an innovative teaching and learning strategy and other appropriate activities, the authors hope to foster students' CT skills, reduce cognitive load, and promote strategy use and sustainable learning.

References

Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference. *Future of Learning Group Publication*, *5*(3), 438. Retrieved from: http://www.sylviastipich.com/wp-content/uploads/2015/04/Coursera-Piaget-_-Papert.pdf

Ali, W. N. A. W., & Yahaya, W. A. J. W. (2020). A Systematic Review of Learning Theory on Computational Thinking. *Journal of Human Development and Communication*, *9*(1), 11–22. Retrieved from: https://www.researchgate.net/profile/Wan-Nor-Ashiqin-Wan-Ali/publication/349074436_A_Systematic_Review_of_Learning_Theory_on_Computational_Thinking/links/601e4d3a92851c4ed54fa95f/A-Systematic-Review-of-Learning-Theory-on-Computational-Thinking.pdf

Anderson, R. C., & Pearson, P. D. (1988). A schema-theoretic view of basic processes in reading comprehension. In P.L. Carell, J. Devine, & D. E. Eskey (Ed.). *Interactive Approaches to Second Language Reading*, 37–55. Cambridge, England: Cambridge University Press. https://doi.org/10.1017/CBO9781139524513.007

Aslan, O. (2009). *The role of gender and language learning strategies in learning English* [Master thesis, Middle East Technical University]. OpenMETU Academic Archive. https://open.metu.edu.tr/handle/11511/18929

Ausubel, D. P. (1962). A subsumption theory of meaningful verbal learning and retention. *The Journal of General Psychology*, *66*, 213–224. https://doi.org/10.1080/00221309.1962.9711837

Bagui, S., & Earp, R. (2003). *Database design using entity-relationship diagrams*. New York, NY:Auerbach Publications.

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

Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., & Morpurgo, A. (2018). Informatics and Computational Thinking: A Teacher Professional Development Proposal Based on Social-Constructivism. International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (pp. 194–205). Springer, Cham. https://doi.org/10.1007/978-3-030-02750-6_15

Blasiman, R. N., Dunlosky, J., & Rawson, K. A. (2017). The what, how much, and when of study strategies: Comparing intended versus actual study behaviour. *Memory*, *25*(6), 784–792. https://doi.org/10.1080/09658211.2016.1221974

BMBWF. (2018). *Digitale Grundbildung: Änderung der Verordnung über die Lehrpläne der Neuen Mittelschulen sowie der Verordnung über die Lehrpläne der allgemeinbildenden höheren Schulen.* Bundesministerium für Bildung, Wissenschaft und Forschung. Retrieved from: https://www.ris.bka.gv.at/Dokumente/BgblAuth/BGBLA_2018_II_71/BGBLA_2018_II_71.html

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education. *European Commission, JRC Science for Policy Report*, *68*. Retrieved from: https://komenskypost.nl/wp-content/uploads/2017/01/jrc104188_computhinkreport.pdf

Chen, P. P.-S. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS)*, *1*(1), 9–36. https://doi.org/10.1145/320434.320440

Clark, J. M., & Paivio, A. (1991). Dual coding theory and education. *Educational Psychology Review*, *3*(3), 149–210. https://doi.org/10.1007/BF01320076

Curzon, P., Bell, T., WAITE, J., & Dorling, M. (2019). Computational thinking. In S. A. Fincher, & A. V. Robins (Ed.). *The Cambridge Handbook of Computing Education Research*, 513–546. Cambridge, England: Cambridge University Press. https://doi.org/10.1017/9781108654555.018

Demarle-Meusel, H., Rottenhofer, M., Albaner, B., & Sabitzer, B. (2020). Educational Pyramid Scheme – A Sustainable Way Of Bringing Innovations To School. *2020 IEEE Frontiers in Education Conference (FIE).* IEEE. (pp. 1–7). https://doi.org/10.1109/fie44824.2020.9274172

Dexter, D. D., & Hughes, C. A. (2011). Graphic Organizers and Students with Learning Disabilities: A Meta-Analysis. *Learning Disability Quarterly*, *34*(1), 51–72. https://doi.org/10.1177/073194871103400104

Ertmer, P. A., & Newby, T. J. (2013). Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective. *Performance Improvement Quarterly*, *26*(2), 43–71. https://doi.org/10.1002/piq.21143

Grabmeier, J. (2018, May 31). *This is your brain detecting patterns: It is different from other kinds of learning, study shows.* Ohio State University. Retrieved from: https://www.sciencedaily.com/releases/2018/05/180531114642.htm

Hall, T., & Strangman, N. (2002). Graphic organizers. *National Center on Accessing the General Curriculum*, 1–8. Retrieved from: https://www.northernhighlands.org/cms/lib5/NJ01000179/Centricity/Domain/18/Graphic_Organizers_2008.pdf

Hsu, T.-C., & Liang, Y.-S. (2021). Simultaneously improving computational thinking and foreign language learning: Interdisciplinary media with plugged and unplugged approaches. *Journal of Educational Computing Research*, *59*(6), 1184–1207. https://doi.org/10.1177/0735633121992480

Kim, A.-H., Vaughn, S., Wanzek, J., & Wei, S. (2004). Graphic organizers and their effects on the reading comprehension of students with LD: A synthesis of research. *Journal of Learning Disabilities*, *37*(2), 105–118. https://doi.org/10.1177/00222194040370020201

Kornell, N., & Bjork, R. A. (2007). The promise and perils of self-regulated study. *Psychonomic Bulletin & Review*, *14*(2), 219–224. https://doi.org/10.3758/BF03194055

Lu, J. J., & Fletcher, G. H. (2009). Thinking about computational thinking. *Proceedings of the 40th ACM technical symposium on Computer science education (pp. 260-264).* https://doi.org/10.1145/1508865.1508959

Martin, P.-Y., & Nicolaisen, T. (2015). *Lernstrategien fördern: Modelle und Praxisszenarien*. Weinheim, Germany: Beltz Juventa.

Morehead, K., Rhodes, M. G., & DeLozier, S. (2016). Instructor and student knowledge of study strategies. *Memory*, *24*(2), 257–271. https://doi.org/10.1080/09658211.2014.1001992

National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. Washington, DC: National Academies Press.

Oxford, R. L. (1993). Instructional implications of gender differences in second/foreign language (L2) learning styles and strategies. *Applied Language Learning*, *4*(1-2), 65–94.

Papert, S. (1980). *Children, computers, and powerful ideas.* United Kingdom: Harvester Press

Parsazadeh, N., Cheng, P.-Y., Wu, T.-T., & Huang, Y.-M. (2021). Integrating computational thinking concept into digital storytelling to improve learners' motivation and performance. *Journal of Educational Computing Research*, *59*(3), 470–495. https://doi.org/10.1177/0735633120967315

Rahmat, N. H. (2020). Information Processing as Learning Strategy: The Case of Graphic Organisers. *European Journal of Education Studies*, *0*(0). https://doi.org/10.46827/ejes.v0i0.3023

Rottenhofer, M., Sabitzer, B., & Rankin, T. (2021). Developing Computational Thinking Skills Through Modeling in Language Lessons. *Open Education Studies*, *3*(1), 17–25. https://doi.org/10.1515/edu-2020-0138

Sabitzer, B., & Pasterk, S. (2015). Modeling: A computer science concept for general education. *2015 IEEE Frontiers in Education Conference (FIE).* IEEE. https://doi.org/10.1109/fie.2015.7344062

Schneider, E., & Crombie, M. (2012). *Dyslexia and Modern Foreign Languages* (0th ed.). Taylor and Francis. https://doi.org/10.4324/9780203458709

Seidl, M., Scholz, M., Huemer, C., & Kappel, G. (2015). *UML@ classroom*. Switzerland: Springer Cham. https://doi.org/10.1007/978-3-319-12742-2

Sousa, D. A. (2017). *How the brain learns* (Fifth edition). London, United Kingdom: Sage Publishing Company.

Susser, J. A., & McCabe, J. (2013). From the lab to the dorm room: metacognitive awareness and use of spaced study. *Instructional Science*, *41*(2), 345–363. https://doi.org/10.1007/s11251-012-9231-8

Sweller, J., van Merrienboer, J. J. G., & Paas, Fred G. W. C. (1998). Cognitive Architecture and Instructional Design. *Educational Psychology Review*, *10*(3), 251–296. https://doi.org/10.1023/A:1022193728205

Taraban, R., Maki, W. S., & Rynearson, K. (1999). Measuring study time distributions: Implications for designing computer-based courses. *Behavior Research Methods, Instruments, & Computers : A Journal of the Psychonomic Society, Inc*, *31*(2), 263–269. https://doi.org/10.3758/BF03207718

Tercanlioglu, L. (2004). Exploring gender effect on adult foreign language learning strategies. *Issues in Educational Research*, *14*(2), 181–193. Retrieved from: http://www.iier.org.au/iier14/tercanlioglu.html

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. https://doi.org/10.1007/s10639-015-9412-6

Wild, K.-P., & Schiefele, U. (1994). Lernstrategien im Studium: Ergebnisse zur Faktorenstruktur und Reliabilität eines neuen Fragebogens. *Zeitschrift Für Differentielle Und Diagnostische Psychologie*, *15*(4), 185–200. Retrieved from: https://psycnet.apa.org/record/1996-85746-001

Willis, J. (2007). Review of Research: Brain-Based Teaching Strategies for Improving Students' Memory, Learning, and Test-Taking Success. *Childhood Education*, *83*(5), 310–315. https://doi.org/10.1080/00094056.2007.10522940

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

World Economic Forum. (2020). *The Future of Jobs Report 2020*. Retrieved from: https://www.voced.edu.au/content/ngv:88417

Yin, R. K. (2009). *Case study research: Design and methods* (4th ed.). *Applied Social Research Methods: Vol. 5*. Thousand Oaks, CA: SAGE Publications

Corresponding author: Mag. Marina Rottenhofer
Email: marina.rottenhofer@jku.at