Research Article

Marina Rottenhofer*, Barbara Sabitzer, Thomas Rankin

Developing Computational Thinking Skills Through Modeling in Language Lessons

<https://doi.org/10.1515/edu-2020-0138>

received July 30, 2020; accepted December 30, 2020.

Abstract: Technology is rapidly changing the world around us and thus, there is a need to adjust education by teaching children skills that are required in the fast-paced digital life. One problem-solving skillset, which has gained considerable attention in the last couple of years, is computational thinking (CT). Up to now, many countries have already implemented CT as an integral part of their education curricula, however, there is still often the misconception that teaching CT requires high technical effort and profound knowledge of computer science. Whereas CT is useful in any subject, it is not necessarily linked to technology and helps children to tackle problems by applying skills that are used in computer science. One effective hands-on approach to foster CT in every subject is modeling. A model is a simplified and reduced version of the real world and modeling is the process of creating it. In this paper, the authors focus on fostering CT skills with models from the field of computer science (CS) in foreign language teaching. The authors present several CS models, that have proven to be useful in language teaching, demonstrate how this approach can foster CT skills and give an insight into their research.

Keywords: Computational thinking; modeling; UML; concept maps; foreign language teaching.

1 Introduction

21st century teachers are facing fundamental educational challenges and need to comply with the changes that

occur at an ever-increasing pace to prepare the pupils of today for the modern job market. The rapid development of technology and new professions that will be demanded soon, requires teachers to educate pupils for an uncertain future. 21st century education is shifting to a competence orientated approach, fostering skills that will help them in the future. One of those skillsets, which has gained increasing attention throughout the years and found its place in many national curricula is computational thinking (CT) (Wing, 2006). The term “computational” itself, with its many different definitions, often lets teachers think that they need to have programming skills or be proficient in the handling of different kinds of technology to be able to implement CT in their subjects. However, fostering CT is not necessarily linked to technology and does not seek to develop IT specialists. It is about cultivating a set of skills that helps people to solve problems and address tasks systematically and efficiently (Barr, Harrison, & Conery, 2011). In the same way as basic language skills help people to communicate, basic CT skills help them to process information and tasks (Lu & Fletcher, 2009). The authors’ aim is to introduce modeling as a tool to teach CT unplugged, without the need of technological devices and to demonstrate to teachers the usefulness of fostering pupils’ CT skills.

A model is an abstract description of a real system that contains the essential elements of this system (Hubwieser, Mühling, & Aiglstorfer, 2015). Models are a vital part of every science and can be categorized as follows: mental (imagination in one’s head), verbal (oral description), graphic (e.g., images, diagrams...), physical (e.g., a miniature house) or formal models (e.g., a computer program) (Fleischmann, Oppl, Schmidt, & Stary, 2018). For this study, the authors focus on graphic models. More precisely, the authors seek to extend modeling with diagrams from the field of computer science (CS) to other subjects by using it as an effective teaching and learning tool. In CS, the Unified Modeling Language (UML) and other models, such as the entity-relationship diagram or graphs, are commonly used to visualize and solve complex problems (Seidl, Brandsteidl, Huemer, & Kappel,

*Corresponding author: Marina Rottenhofer , Department of STEM Education, Johannes Kepler University, 4040 Linz, Austria, E-mail: marina.rottenhofer@jku.at;

Barbara Sabitzer , Department of STEM Education, Johannes Kepler University, 4040 Linz, Austria

Thomas Rankin , Center for Business Language & Intercultural Communication, Johannes Kepler University, 4040 Linz, Austria

2012). Also, outside the field of CS, these diagrams have many application possibilities and can be compared to concept maps, which are known to have major effects on students learning outcomes, especially on students with learning difficulties (Sousa, 2016; Knogler, Wiesbeck, & CHU Research Group, 2018).

In this paper, the authors focus on modeling in foreign language teaching as a new approach to teach CT without the use of technological devices. Besides best practice examples, the paper presents some results gained throughout the years and gives an outlook on future investigations.

2 Related Work

Since 2006, when Jeanette Wing (2006) introduced computational thinking (CT) as a fundamental skill for everyone, such as reading, writing, or arithmetic, much research has been going on and numerous definitions of CT emerged. According to Wing, CT includes mental processes and concepts that are independent of technology and used to deal with problems in the field of computer science. Furthermore, she refers to the representation of problems by stating that CT also stands for “modeling the relevant aspects of a problem to make it tractable” (Wing, 2006), which underlines the suitability of modeling as a tool to foster CT. Mindfully designing models requires to deeply engage with the learning content that is supported through CT, which according to the Computing at School Association (Csizmadia, Curzon, Dorling, Humphreys, Ng, Selby, & Woollard, 2015) focuses on the thought process, supports learning and understanding and “allows pupils to tackle problems, to break them down into solvable chunks and to devise algorithms to solve them”.

The literature on CT focuses on different key concepts that support general learning and understanding in a range of areas. With modeling as a hands-on approach to teach CT, the authors seek to foster the core CT skills which are presented by the Joint Research Center (JRC) such as abstraction, algorithmic thinking, automation, decomposition, debugging and generalization (Bocconi, Chiocciariello, Dettori, Ferrari, Engelhardt, Kampylis, & Punie, 2016).

Computational thinking as a problem-solving process and modeling as a visualization tool can support the demanding processes that occur in foreign language learning. Language is an infinite system and so, by definition, children learning their native language and older children or adults learning a subsequent language

can only be exposed to a limited range of instances of linguistic performance (Newport, 1990). From this limited corpus of linguistic structures, learners are required to deduce the underlying grammatical rules which generate the full range of structures that a language allows.

According to the literature, there have already been attempts to implement CT in language lessons. Barr and Stephenson (2011) for instance, present how CT concepts can be embedded in activities such as linguistic analysis of sentences, identification and representation of different patterns for different sentence types, writing an outline, using simile and metaphors, story writing with branches, writing instructions, etc. Also, Lu and Fletcher (2009) demonstrate several examples of the use of a CTL (Computational Thinking Language) in language arts such as applying computer science methods like *divide-and-conquer* or *pruning* for reading comprehension or *recursion* and *non-determinism* for grammar. The proposed approaches are ideal when students encounter more complex situations and help to process information and tasks more systematically and efficiently. However, methods such as recursion or non-determinism require profound knowledge in computer science and can be therefore very demanding for teachers with no CS background. With modeling, the authors seek to span the bridge between CS and other subjects and aim to eliminate the teachers' fears.

Originally, the aim of graphic modeling in the field of computer science was to facilitate discussions about software design and in 1997, UML was born to unify the many modeling languages that boomed in the late 80s and early 90s (Fowler, 2004). Due to its ability to extract the essentials of a complex system and visualize situations, states, processes, relations, or hierarchies, modeling is also an effective method in other disciplines that involve complex systems. Eriksson and Penker (2000) share the same opinion by using UML for business modeling. To the best of the authors' knowledge, in the field of education, UML models are solely used in the context of computer science. However, the authors claim that UML models are a very effective teaching and learning tool for any subject, and they are easy to acquire and implement. Furthermore, it represents an opportunity for cross-curricular cooperation between computer science and other disciplines.

The following section focuses on the use of modeling in foreign language learning and represents four models, that have proven to be very suitable to implement CT in the language classroom.

3 Modeling in Language Teaching

The Unified Modeling Language (UML) helps students to master language learning with confidence providing a wide range of diagrams that are suitable for all levels of language complexity. Besides UML, other models from the field of computer science, such as the entity-relationship model, can be effectively used in different areas of language learning such as grammar, vocabulary learning, writing, reading, or speaking.

One major area of foreign language learning students often struggle with is grammar. When it comes to grammar teaching, there are many different types: pedagogical, reference, prescriptive, linguistic grammar, etc. and each of these has different potential advantages and disadvantages for language teaching and learning (Larsen-Freeman, 2011). By implementing a modeling approach, the complexity of grammatical learning can be adapted for different learning needs. For example, relatively simple grammatical rules of thumb can be visualised and used to promote pattern recognition, which would be useful for less complex features and at lower levels of language proficiency. However, at higher levels of proficiency or for more intricate areas of grammar, the use of algorithmic thinking could be employed in a more exploratory way to allow deductive decomposition of complex usages into useful rules. Similar approaches to modeling are widely used in the linguistic study of grammar within different formalisms, and ideas such as sentence diagramming were for long a traditional aspect of grammar teaching in both L1 and L2 contexts. However, a diagramming approach seems to have fallen into disrepute in more recent approaches to foreign language teaching, perhaps due to associations with a rigid and ‘old-fashioned’ grammar-translation approach to language teaching. Nevertheless, it would be advantageous if the best of the rationale for diagramming (visualisation, clarification) is combined with the rationale of deductive learning and achieving cross-curricular goals to promote autonomous and productive language learning.

A modeling approach can be viewed as an antidote to the use of long and complex explanations of grammatical rules or unrelated lists of vocabulary. While there is nothing inherently wrong in such explanations or lists, they may not be accessible for learners and so hinder engagement with language. Pupils are not conceptualised as passive learners sitting in front of their textbooks, reading grammatical explanations or learning long lists of vocabulary and trying to remember all the rules, words and exceptions to be finally able to put the knowledge gained

into practice. Rather, learners should be engaged in trying to figure out the nature of the rules within a framework that promotes the abilities to decompose data into useful patterns and abstract away from the data to form rules and promote pattern recognition. Further, learners should deeply engage with the vocabulary instead of learning them by rote, by clustering the words or putting them in a context.

Besides learning vocabulary and grammatical rules, modeling serves as an intermediate step when working with texts. When pupils have difficulties in extracting the essential information of texts or understanding the meaning, modeling allows them to decompose the text in small parts, abstract essential information and recognize patterns and relations and thus, promotes successful reading comprehension and summary writing. Furthermore, creative tasks such as role-plays, and other speaking activities or creative writing are well served by modeling. The following section presents several models, that have proven to be very suitable for different contexts of language learning.

4 The Models

Generally, diagrams from the field of computer science are divided into static and dynamic diagrams. For the use of CS modeling as a teaching and learning tool in other subject areas, the authors analysed many different diagrams and developed a categorization, which can be linked to any subject (see figure 1).

In the context of language learning, diagrams from all three categories have proven to be easily implemented and useful in different areas of language learning. Class & object diagrams, for example, are very useful when it comes to categorize and cluster vocabulary. Major findings reveal that the thematic clustering of L2 vocabulary facilitates learning of new words (Tinkham, 1997). According to the authors’ experience teachers are often surprised how many CT concepts they already implement in their classroom unconsciously and how much CS there is in language teaching, without even using a computer. One example is the graph, which is frequently used by language teachers (e.g., metro map). As can be seen in figure 1, dynamic diagrams, such as a flowchart or activity diagram are very suitable to represent rules and procedures. Algorithmic thinking and decomposing grammar rules in small chunks may help pupils to put the theoretical knowledge into practice. The third category, situations and states, provides diagrams that are very useful when dealing with complex texts or preparing

Structures & Categories **Rules & Procedures** **Situations & States** **Figure 1:** Modeling categories.

speaking activities by visualizing relations, abstracting the main information, etc.

Before presenting one model for each of the categories, it is worth mentioning how CS diagrams are used in an interdisciplinary context as a teaching and learning tool. The field of computer science has already voiced criticism that the diagrams are often not fully correct according to their standards. Nevertheless, although the diagrams derive from the field of computer science, the primary focus always lies on the successful visualization of the subject matter and not on the syntax of the diagrams. In other words, all the characteristics of the individual diagrams, such as the correct notation of attributes, methods, etc. must not necessarily be fulfilled. For example, the name, attributes and methods of a class can also be referred to as nouns, adjectives or verbs if this serves the purpose of the exercise. The authors see modeling as a tool to trigger deep thinking processes and therefore, a too strong focus on correct syntax from the computer science perspective could have negative effects by discouraging students and teachers. Even though the correct syntax is not the core focus of modeling, the question arises as to when a model can be declared as UML or other computer science model and not just as random visualization. To answer this question, we have developed an assessment tool called the *Reference Framework of Modeling (ReMo)* (Sabitzer, Demarle-Meusel, & Rottenhofer, 2020), where stakeholders can rate their modeling proficiency and receive information about which mental processes are happening when creating a model.

4.1 Class and Object Diagram

Class and object diagrams are both UML models and used to visualize structures and categories. More precisely, the

class diagram is used to visualize the different elements of a system and how they relate to each other. It is one of the most popular UML diagrams and, due to its simplicity, widely used for visualizing the classes of a software system and the relationships between them. The object diagram, on the other hand, represents instances of the class diagram (Seidl et al., 2012).

Class and object diagrams are easy to model. They are visualized with rectangles and divided into several compartments. The first compartment of the class diagram contains the name of the class. The second compartment contains the attributes, and the third one, the methods or operations. Different types of relationships can be visualized between the single classes, as, for example, aggregation, association, or generalization/inheritance.

Figure 2 represents a generalization or inheritance of classes. The generalization relationship can be used to represent classes that have attributes and/or methods in common and is indicated with a blank arrow leading from the sub-class to the more general class. As illustrated in figure 2, the classes “teacher” and “student” belong both to the category “person”, and also share the attributes and methods of the class “person”. The classes “teacher” and “student” inherit these characteristics and therefore, there is no need to mention them again. To the inherited characteristics, individual ones can be added to the subclasses (e.g., the methods “teach” and “study”).

Object diagrams are used to visualise concrete objects of a system and the link between each other. The classes of a class diagram are used as templates for the concrete objects. In other words, all the attributes and operations of the classes are specified in the objects (Seidl et al., 2012).

Figure 3 illustrates an object diagram with two objects. *Ms. Cooper* is the teacher of *Thomas* and these two objects are specified according to the attributes and operations of the classes “teacher” and “student” in figure 2.

Figure 2: Class diagram and inheritance.

Figure 3: Object diagram.

Figure 4: Sequence of events.

4.2 Activity Diagram

As the name already indicates, an activity diagram is used to represent an activity. More precisely, it is used to visualize the single steps of an activity. In language teaching, grammar rules, processes or events of a story are some examples of activities. In the designing process of this model, several steps are required. One of them is algorithmic thinking, which is one of the crucial components of computational thinking and implies the precise definition of individual steps in a problem-solving process. However, before students can visualize their algorithms, they need to be familiarized with the most important elements for drawing activity diagrams.

Experience shows that the following components and rules are sufficient for the first encounter with this form of modeling.

The main elements of the activity diagrams are rounded rectangles that represent the *single actions*. To illustrate, figure 4 shows an activity diagram with single steps of a grammar task. As in this example, the actions of a process always lead from a clearly defined starting point to an endpoint which are called *initial and final node*.

The single actions in the context of an activity are always seen as atomic. In other words, in the modeled activity, these actions cannot be further broken down (Seidl et al., 2012). As an example, figure 4 represents the process of a grammar task a teacher gives to her students. “Read grammar explanation”, “draw an activity diagram”, and “fill in the gaps” are single actions of the activity “doing a grammar task”. In this process, they are seen as atomic - as the smallest particles. However, one of these actions can refer to another activity that contains several individual steps. Figure 4 considers “draw an activity diagram” as one action. If you think about this element as an activity rather than an action, it becomes evident that drawing the diagram requires multiple steps such as “extracting main information from the text”, “taking pen and paper”, “drawing shapes”, etc. For the model in figure 4, however, the procedure of putting the diagram onto paper is not of relevance and therefore, seen as a single action. To summarize, given the divisibility of the actions, all the single steps of an algorithm are represented separately in rounded rectangles.

An activity that follows another is a *sequence* and is connected with *edges* (arrow or control flow) that indicate the reading direction. To visualize a decision (figure 5) a diamond shape is used as a *decision node* or *conditional branch*, which always includes at least two different control flows.

If you want to repeat an action until a certain condition is met, you can visualize that with a loop. In figure 6, the student has to repeat the action of “filling in the gaps” until all the sentences are completed. If that is the case, the student can leave the loop and continue to the next action.

4.3 Entity-Relationship Diagram

The entity-relationship (ER) diagram is a static model, which does not belong to the family of UML models. For using it as a teaching and learning tool, the authors refer to the “Chen notation” (Chen, 1977) which consists only of three elements and is thus easy to acquire and suitable

for all levels. These three elements are called *entity types*, *attributes* and *relationship types*. Entity types describe a group of real objects and are represented in rectangles. Similar to a class, the entity does not have a specific, but a generic name that represents a type of a thing rather than an instance (Bagui, & Earp, 2011).

The characteristics of the entity types are called attributes and visualized as ellipses. The link between the entity types are represented in diamond shapes and called relationship types. All the elements are connected with simple lines.

Figure 5: Decision node of an activity diagram.

Figure 6: Looping in an activity diagram.

As an example, figure 7 represents the entities “school”, “student” and “grammar task” with its attributes in ellipses. The relationships between these entities are “goes to” and “receives”. It is essential for this diagram, that the entities and attributes are represented as abstract terms (e.g., student instead of Thomas or gender instead of male).

In the language classroom, ER diagrams are useful when working with texts, because it encourages noticing. In other words, learners focus on the language (not on the content) of a text, try to recognize patterns and subsequently create a model. Since the ER diagram uses generic terms, generalisation too can be trained with these diagrams. However, used as a teaching and learning tool, the diagram syntax can be handled flexibly. In other words, if a task requires the visualisation of specific terms, then this diagram can also be used in a modified form. With this flexible application, the ER diagram is a versatile method when working with texts.

5 Methods and Results

In the last couple of years, modeling as (1) an interdisciplinary teaching and learning tool (Sabitzer, & Pasterk, 2015) and (2) a tool especially for foreign language learning (Sabitzer, Demarle-Meusel, & Jarnig, 2018) has always been a focus of attention. Several projects, where pupils, students and teachers participated in workshops, talks or training sessions dealing with (interdisciplinary) computer science at school, underlined its effectiveness. Throughout the years, the authors especially noticed the recurring positive feedback of language teachers and decided to specifically investigate the use of modeling in this subject. In 2014, one project already focused on the use of diagrams for text work in foreign language lessons.

Figure 7: Entity-relationship diagram.

Figure 8: The city tour bus.

The study of 2014 involved 141 students and revealed that modeling especially encourages students to filter out essential information of texts, which helps them in their writing process (Salbrechter, Kölblinger, & Sabitzer, 2015). In this empirical study, modeling was proven to be useful for language teaching, especially when it comes to recognizing essential information, which is also crucial in grammar teaching. The following figure (8) illustrates a best practice example for lower-intermediate level pupils. In this exercise, pupils have the task to identify parts of speech by analyzing the words in the shapes. The follow-up activity shows an incomplete ER diagram and the generalized terms of the text above which must be matched with the blank shapes. At a more advanced level, pupils can use the highlighted parts of speech to model their own diagram with concrete and/or generalized terms.

In 2018, we organized a teacher training workshop for language teachers. The workshop lasted two days and involved 13 language teachers, of whom one was male. On the first day, a questionnaire was given to the teachers, to find out whether they are familiar with modeling and CT and other visualization strategies. Only one person (7.7%) knew modeling before the workshop and only two (15.4%) were familiar with the concept of computational thinking. That is not very satisfying, since Austria introduced the new curriculum “Basic Digital Education” (BMBWF, 2018), with CT as a crucial part of it, in the same year.

When asking about other visualization strategies, the questionnaire revealed, that most teachers (n= 10) know mind-maps, but only two of them concept maps, which can be compared to modeling. Throughout the workshop, the teachers were introduced to CT as an important problem-solving skill and five different models as tools to implement it. After the introduction, teachers worked on their own teaching materials and prepared activities, they could immediately implement in their lessons. Developing their own materials required them to deeply engage with the models in an environment, where 3 experts provided continuous support. Explicit examples of materials developed by the teachers would be the representation of different text types with class and object diagrams or a step by step instruction for a grammar task represented with an activity diagram. On day two, the teachers presented their material to the group followed by a discussion and feedback session. At the end of the workshop, the teachers were asked to complete a second questionnaire to find out more about their perception of modeling. The majority of the teachers (n= 12) are convinced of the effectiveness of modeling and will use some of the diagrams in future lessons. Concerning the advantages and implementation possibilities, the answers can be summarized as follows:

- Advantages:
 - Logical and clear structure and overview
 - Useful learning tool which reaches different learning types

- Varied, motivating and content easier to remember
- Implementation:
 - Reading comprehension, structuring and summarizing texts, grammar and vocabulary learning, speaking activities

The teachers were also asked about whether they see any disadvantages or difficulties when implementing modeling. Twelve teachers share the same opinion by claiming that there might not be enough time to use it in the classroom because it is very time-consuming. However, through verbal feedback and observations during the workshop, we found out that many teachers question the correctness of their models from a computer science point of view. They worry, that their version might just be a blueprint, that is not suitable for promoting CT. This was perhaps one of the reasons for the large amount of time required for creating the models. These findings revealed the importance to clearly show teachers that the focus of modeling as a teaching and learning tool is not the correct syntax from a CS perspective, but the thinking processes that occur when creating a model. Even though the model looks like a blueprint, skills such as abstraction, algorithmic thinking, pattern recognition, generalization, etc. must be applied. To make this assumption clearer, we developed the *Reference Framework of Modeling (ReMo)*, Sabitzer, Demarle-Meusel, & Rottenhofer, 2020) which is sparked by the *Common European Framework of Reference for Languages (CEFR)*, Council of Europe, 2001) and serves as a modeling assessment tool.

Shortly after the workshop with language teachers, the Erasmus+ project *Modeling at School* (2018-2021) initiated, where we work with many different partner schools in Austria, Finland, and Spain and spread the concept of modeling with a novel training method called *Educational Pyramid Scheme (EPS)*, Demarle-Meusel, Rottenhofer, Albaner, & Sabitzer, 2020), which involves teachers and students who collaborate to effectively implement innovations at school. Focus interviews held in 2020 with teachers and students of the Austrian partner schools again underlined the usefulness of modeling in language learning with a particular focus on the effectiveness of activity diagrams and algorithms in grammar teaching. According to language teachers, with modeling, grammar structures can be clearly represented and especially in this area the pupils were very enthusiastic about this type of representation because the individual steps became clear to them.

6 Conclusion and Outlook

In conclusion, the field of computer science offers a variety of diagrams (mostly UML), which are not only useful to solve complex CS problems but also serve as an effective teaching and learning tool. The process of modeling requires skills such as decomposition, abstraction, pattern recognition, algorithmic thinking, etc., which are essential components of computational thinking and require pupils to deeply engage with the learning content. Especially in foreign language teaching, modeling can serve as a bridge between this subject and computer science and help to understand and master a foreign language with confidence and inspire pupils, especially girls, for the STEM (science, technology, engineering and mathematics) field. The authors see modeling as a tool to trigger deep thinking processes and aim at bringing this innovation into schools with the *Educational Pyramid Scheme*, which involves teachers and pupils in the training process. The data gained throughout the years has shown promising results and underlines the potential of modeling in foreign language teaching and learning. The next step is to delve deeper into the subject matter by investigating its effects on vocabulary and grammar acquisition and the thinking processes that occur when pupils develop different models.

References

- Bagui, S., & Earp, R. (2011). *Database design using entity-relationship diagrams*. Crc Press.
- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23.
- 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.
- BMBWF Bundesministerium für Bildung, W. u. F. (2018). Digitale Grundbildung. In: Änderung der Verordnung über die Lehrpläne der Neuen Mittelschulen sowie der Verordnung über die Lehrpläne der allgemeinbildenden höheren Schulen. Retrieved from: https://www.ris.bka.gv.at/Dokumente/BgblAuth/BGBLA_2018_II_71/BGBLA_2018_II_71.pdf
- Bocconi, S., Chiocciariello, 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*. Luxembourg, Publications Office of the European Union.
- Chen, P. P. S. (1977, June). The entity-relationship model: a basis for the enterprise view of data. In *Proceedings of the June 13-16, 1977, National Computer Conference* (pp. 77-84).
- Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division. (2001). *Common*

- European Framework of Reference for Languages: learning, teaching, assessment.* Cambridge University Press.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking-A guide for teachers.* Swindon: Computing at School.
- Demarle-Meusel, H., Rottenhofer, M., Albaner, B. & Sabitzer, B. (2020). Educational pyramid scheme – a sustainable way of bringing innovation to schools. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1-7). IEEE.
- Eriksson, H. E., & Penker, M. (2000). *Business Modeling with UML.* New York: John Wiley & Sons, Inc.
- Fleischmann, A., Oppl, S., Schmidt, W., & Stary, C. (2018). Modelle. In *Ganzheitliche Digitalisierung von Prozessen: Perspektivenwechsel–Design Thinking–Wertgeleitete Interaktion* (pp. 19-69). Wiesbaden: Springer Vieweg.
- Fowler, M. (2004). *UML distilled: a brief guide to the standard object modeling language.* Boston: Pearson Education, Inc.
- Hubwieser, P., Mühlh, A., & Aiglstorfer, G. (2015). *Fundamente der Informatik: Funktionale, imperative und objektorientierte Sicht, Algorithmen und Datenstrukturen.* Oldenbourg: Walter de Gruyter.
- Knogler, M., Wiesbeck A. B. & CHU Research Group (2018). Lernen mit Concept Maps: Eine Bilanz nach 42 Jahren Forschung. Retrieved from: www.clearinghouse-unterricht.de, Kurzreview 19.Larsen-
- Larsen-Freeman, D. (2011). Teaching and testing grammar. In M. Long and C. Doughty, (Eds.,) *The Handbook of Language Teaching.* (pp. 518-542) Oxford: Blackwell Publishers.
- Lu, J. J., & Fletcher, G. H. (2009). Thinking about computational thinking. In *Proceedings of the 40th ACM technical symposium on Computer science education* (pp. 260-264).
- Newport, E. L. (1990). Maturation constraints on language learning. In *Cognitive science*, 14(1) (pp. 11-28).
- Sabitzer, B., Demarle-Meusel, H., & Jarnig, M. (2018, April). Computational thinking through modeling in language lessons. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1913-1919). IEEE.
- Sabitzer, B., Demarle-Meusel, H., & Rottenhofer, M. (2020, February). Modeling as Computational Thinking Language: Developing a Reference Framework. In *Proceedings of the 2020 9th International Conference on Educational and Information Technology* (pp. 211-214).
- Sabitzer, B., & Pasterk, S. (2015, October). Modeling: A computer science concept for general education. In *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1-5). IEEE.
- Salbrechter, C., Kölblinger, I., & Sabitzer, B. (2015, March). Modeling – a computational thinking concept and tool for cross-curricular teaching. In *INTED2015 Proceedings, ser. 9th International Technology, Education and Development Conference. IATED*, (pp.4280–4290)
- Seidl, M., Brandsteidl, M., Huemer, C., & Kappel, G. (2012). *UML@ Classroom: Eine Einführung in die objektorientierte Modellierung.* Heidelberg: dpunkt. verlag.
- Sousa, D. A. (2016). *How the brain learns.* Thousand Oaks: Corwin Press.
- Tinkham, T. (1997). The effects of semantic and thematic clustering on the learning of second language vocabulary. *Second language research*, 13(2), 138-163.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.